FEATURE BASED DESIGN THROUGH REVERSE ENGINEERING

by

SWAPNIL BHANUDAS CHANDSARKAR

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

March, 2000

FEATURE BASED DESIGN THROUGH REVERSE ENGINEERING

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

SWAPNIL BHANUDAS CHANDSARKAR

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR – 208016 (INDIA)
March, 2000

CERTIFICATE

It is certified that the work contained in the thesis entitled, "FEATURE BASED DESIGN THROUGH REVERSE ENGINEERING" by *Mr. Swapnil Bhanudas Chandsarkar* has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

8-3-2000

Dr. S. G. Dhande.
Professor
Department of Mechanical Engineering,
Indian Institute of Technology, Kanpur.

Dedicated

To

My Parents

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude to my ever-cherished guide, Dr. S.G.Dhande for his invaluable guidance and help throughout my M. Tech. programme. I am sincerely thankful for his valuable suggestion in my academic as well as personal life. He has been and would be a constant source of encouragement to me throughout my life, particularly for his kind and forever-calm nature even in tough situations. I am paying great regards to him for giving me an opportunity to work in leading areas of CAD, Reverse Engineering, RP & RT. He was and continues to be a model of dedication, discipline, and hard work, which was a great inspiration to me.

I am grateful to Shri Puneet Tandon and Rahul Kumar for the useful discussion I had with him regarding my thesis work. My sincere thanks and regards to Santosh Kulkarni, for the technical discussion I had with him, his motivations and advice. Thanks to Sanat*ji*, Mukul*ji*, Siva Prasad*ji* and Sarvanakumar*ji* for suggestion and support.

I am thankful to my CAD-P friends Saurabh, Siddarth, Deepak, Suresh, Rajil, and Anil. My sincere thanks to Yogesh Kulkarni for his throughout assistance, amity and company.

Goods friends make the life enjoyable. I wish to thank to all my *Ghatmandal* friends, who shared and stood by me during days of happiness as well as difficulties, I am thankful.

I would like to thank to all CAD–P Lab, Shri C. P. Singh, Shri S. C. Gupta, Shri Ravindra Srivastava, Shri BabuLal, Shri Sarju Prasad and others for their helps in one way or others.

I wish to express my heartfelt thanks to my parents and elder brother, for their endless love, encouragement and endurance during my stay at I. I. T. Kanpur.

Finally, I am grateful to the Almighty for what I am today.

Swapnil Chandsarkar

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The objective of present work is to develop a methodology of geometrical design of a component based on a set of specific features. Feature based design, as it is called can be involved when a product is to be designed based on the point cloud data of a model of the component. The acquisition and processing of point cloud data is referred to as the process of reverse engineering. The present work is aimed at evolving a methodology of feature based design using point cloud data and applying it for the design of components in consumer industry.

The proposed methodology consists of creating a geometric model from the point cloud of the component. The process involves definition of curves and surfaces on the point cloud. Using surface definitions, the solid model of the part can be built. During this process, the feature library is developed which is updated as a new feature is encountered. The description of features in this library contains four layers, which has the information of surfaces, curves and feature points. The set of curve features is the core of feature description and is used to develop a particular surface feature. These features are also modified so as to convert them into generic definition. Using the parameterization technique, one can obtain a variety of surface models, which are aesthetically more elegant. Functional intent can also be preserved if required.

In the present work, an attempt has been made to integrate the techniques of reverse engineering and feature-based design. The process of integration is handled in two stages: feature extraction stage and feature based modeling stage. The method developed to extract the features is automatic while the process of feature based modeling is interactive. The overall procedure is explained by taking an example of creation of the CAD model from a real life object.

Chapter 1

INTRODUCTION

1.1 NEED OF FEATURE BASED APPROACH IN CAD

Large multifunctional companies often experience problems because of poor communication between divisions. While product design divisions have sound knowledge about designing products, they have less expertise in areas like manufacturing and maintenance. Companies have begun to investigate ways to conduct an analysis of proposed product designs to eliminate downstream problems of manufacturing, assembly, maintenance, etc. Such design approaches are referred to as Concurrent Design or Concurrent Engineering. Various tools, which have been developed with the help of CAD, are Computer Aided Design for Manufacturing, Computer Aided Design for Assembly, Group Technology, and Computer Aided Design for Process Planning, Reverse Engineering, etc.

A feature based representation of a new or in-progress product designs is an essential prerequisite to development of a new generation of more intelligent computer-aided design systems that can support conceptual design, offer manufacturability advice, provide automatic access to analysis procedures, and capture and use certain designer intentions related to function and manufacturability. Feature based representations can be obtained by feature extraction from existing representations, or by designing with features.

1.2 REVERSE ENGINEERING CONCEPTS

Reverse Engineering refers to an approach for creating a CAD model from an existing physical object whose geometrical or technical information is partially or completely unavailable in digital form.

Reverse engineering process is carried out in three basic phases [Fig. 1.1].

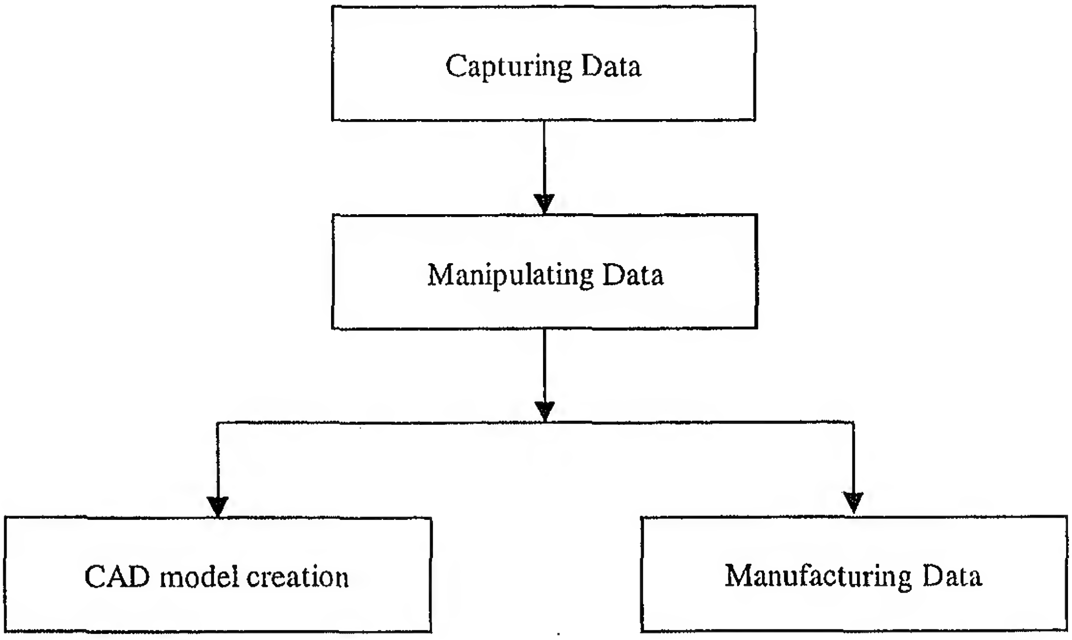


Fig. 1.1: Basic Phases of Reverse Engineering

The approach is in contrast to a normal production process starting from design and ending up with a physical prototype or product. It can be utilized for cloning an existing part, modifying an existing design based on physical products, repairing a worn or damaged part, or re-engineering an entire existing system. In addition, reverse engineering techniques are also very useful in the process of new product development.

With today's technology, the creation of a compete digital model can be done using a CAD system in front of a computer monitor. There are, however, limitations in using CAD systems for free-form shape design. Creative designers often prefer to sculpt a physical mockup for designing free-form surfaces. Reverse engineering techniques can then be applied to transfer the geometrical information of the physical mockup into a CAD system. Typical examples can be found in the automobile industry, ship building industry, aircraft industry, toy making industry and many other industrial fields where free-form surfaces are involved.

1.3 PROBLEM DEFINITION

With the introduction of feature modeling, object semantics are systematically represented for a specific application domain. There are two approaches for building a feature model:

1. The design by feature approach creates the feature model of an object by composing the available features in a feature library.
2. The feature recognition approach recognizes various features from a geometric model of an object according to the feature templates defined in a feature library.

This work discusses an approach of recognizing the features of a sculptured object, which is represented by a cloud of points. The recognition process is based on the analysis of the control plot of the interpolated B-spline curve of degree three, fitted at different cross-sections of the point cloud. Parameterization is set up between different features. The feature model is constructed using these features, which is then used to create the surface model for the design of similar product but with different parameters. This approach can be applied to the design of saddletrees in the saddlery.

1.4 ADVANTAGES

The advantages of this method are:

1. Improvement of the efficiency and accuracy. Using the non-contact measuring method gives a more efficient and accurate measurement than the traditional method.
2. Completeness of the information. Not only are the sizes of the object given by this method, but the shapes of the object are also recorded.

1.5 LITERATURE REVIEW

Feature recognition methodologies can be classified into two major categories: surface recognition and volume recognition [1]. The main difference between these two categories is mainly due to the feature definition and object description in the recognition method. The features and object descriptions in the surface recognition category are expressed in terms of a set of faces and edges. The graph-based methods, syntax-based method, rule based method and neural network method are examples of this category. On the contrary, the features and object descriptions are defined in terms of primitives in the volume recognition category. Typical examples are the convex hull algorithm, hint-based reasoning method and curvature region approach. All these methods use solid model of the object to extract the features which consists of surfaces, which can be defined as a set of Boolean operations applied to standard primitives.

In the paradigm of feature-based design, the basic unit is a feature and parts are constructed by a sequence of feature attachment operations. The type and number of possible features involved depend upon product type, the application reasoning process and the level of abstraction. Therefore to provide CAD systems with a basic mechanism to define features that fit the end user needs seems more appropriate than trying to provide a large repertoire of features covering every possible application. Hoffmann C. and Arinyo R. [2] propose a procedural mechanism for generating and deploying user-defined features. In this paper, emphasis is given on parametric design.

In reverse engineering, the geometric model of an object is built from a cloud of points. This method is commonly used in modeling a sculptured object. Sarkar B. and Menq C-H [3] depict the steps involved in the reverse engineering process. The discussion mainly focuses on the parameterization of these unorganized points. These discussions emphasize the division of the whole array of measurement data points into regions, according to shape-change detection and smooth parametric surface approximation by the least-square fitting of B-spline curves in each of these regions. The process is geometry oriented and feature technology is not beneficial to it. In order to create a feature model of an object from a point cloud, the embedded features must be recognized. These features are then used to constrain the fitting process.

Dobson G., Waggenspack W. and Lamousin H. [4] describe the procedure to extract the embedded features from facial profile with the help of a generic model having similar shape. The procedure consists of aligning the model to given data on the basis of the min/max box, principal axes and landmarks and optimizing vertices and weights of the rational B-spline to locally deform the model and approximate the data to within a specified tolerance. The feature based modeling approach is demonstrated in 2D.

Au C.K. and Yuen M.M.F. [5] discuss the issues of applying feature technology to the reverse engineering of a mannequin. The discussion concentrates on recognition of features from the point cloud by comparing with a generic feature model, setting up the association between the point cloud and the mannequin features and optimizing the distance between the point cloud and the feature surface.

1.6 ORGANIZATION OF THESIS

In the present scenario, it is observed that in many industries, designers want to reproduce the part for which drawings are not available. This is where reverse engineering finds an application. In most of the cases, difference in the design of parts is just due the variation of some parameters. In those situations, the usual process of reverse engineering need to be carried out in a systematic way, wherein the underlying features are extracted as an initial step and using those features, the same process for further parts is made quick and easy.

In chapter 1, the need of feature based approach in CAD is emphasized. Various tools that have been used as a part of CAD have been highlighted. Next, reverse engineering concept has been introduced. The problem tackled in this work and its advantages are stated. The work that has been carried out earlier in the area of feature based design and reverse engineering, is depicted at the end.

Chapter 2 introduces the concept of feature. The principles of feature recognition and feature based design are discussed through examples. Although feature based modeling technique is useful, the concept of user-defined features is emerging as a powerful and convenient tool. To use the use-defined features, how to develop the design libraries for CAD is next. Limitations that are found in traditional ways of using features are explained at the end.

Reverse engineering process has been thoroughly discussed with the example of surface model development of a turbine blade in chapter 3. Some light is thrown on the role of reverse engineering in product development cycle. Applications of reverse engineering are pointed out next.

Integration of feature based design and reverse engineering is discussed in chapters 4 and 5. The procedure is discussed with an example In chapter 4, as the initial step, how the curve based features are extracted, is explained. In chapter 5, the steps of surface based feature extraction are discussed. Once the feature model is developed, parameters are associated to it and how the process of reverse engineering is simplified when applied to a similar product, is explained at the end. The complete description of the example is given in section 5.6, the case study, Results are put at the end to check the feasibility of the process.

In the last chapter, conclusions derived from this work and the scope to improve it is discussed.

Chapter 2

FEATURE BASED DESIGN METHOD

2.1 PRIMITIVTES TO FEATURES - A HIERARCHIC APPROACH

There are many published definitions of the concept of a feature. Even though these definitions seem to be dissimilar, they all consider features as entities, which are of semantically higher level than the pure geometric elements typically used in solid modeling systems. Geometric elements are solid primitives in constructive solid geometry (CSG) type solid models (blocks, cylinders, spheres, torii) or boundary elements used in boundary representation (B-rep) type solid models (faces, edges, vertices) [Fig.2.1]. Almost universally, the concept of generic feature classes is used, and models are built from instances of generic features. The generic types may be organized into feature taxonomy, often realized as a collection of classes with inheritance of information according to the principles of object-oriented programming.

In the type-instance approach, feature instances are represented in terms of various feature attributes. Common attributes include the intrinsic geometric attributes of the shape corresponding to the feature (length, width, depth, radius, etc.) the position and orientation of the feature with respect to some global co-ordinate frame, geometric tolerances, material properties, and references to adjacent and other features. Types contain information shared by all instances of the type. In the object-oriented approach to features, this information often is in the form of procedures for computing interesting properties of the instances, such as volume and cost.

A useful categorization of the various definitions is the separation of "top-down" definitions and "bottom-up" definitions. Top-down definitions emphasize

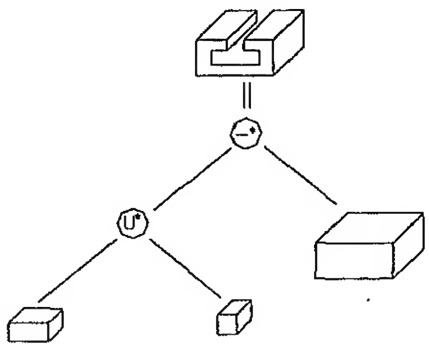


Fig. 2.1: A feature represented in CSG representation

the designer's view of features as elementary entities of a part definition; part geometry is considered as a property of a feature-based part definition, which can be computed on the basis of the feature definition. Bottom-up definitions emphasize features as abstractions of recurring combinations of geometric elements.

The above categorization typically reflects the primary feature creation method used by feature-based modeling systems. Thus, top-down definitions correspond with systems utilizing primarily design-by-features method, where models are originally defined in terms of their constituent features. Similarly, bottom-up definitions correspond with systems utilizing feature recognition, where features are extracted from a previously generated geometric model.

Feature is a concept depending on the context in which a part is considered [Fig. 2.2]. Features can be defined from different viewpoints, such as design, analysis, assembly and function. Because of this, there may be several co-existing feature models of the same product design. This gives rise to the problem of feature mapping, i.e. conversion among the various viewpoints.

A simple object is represented in the top of the figure: a box with two holes which have another hole in their bottom faces, a slot and a pocket on the bottom face of the slot. The shape feature recognizer extracts these six shape features and arranges their primitive volumes in the hierarchical graph representation.

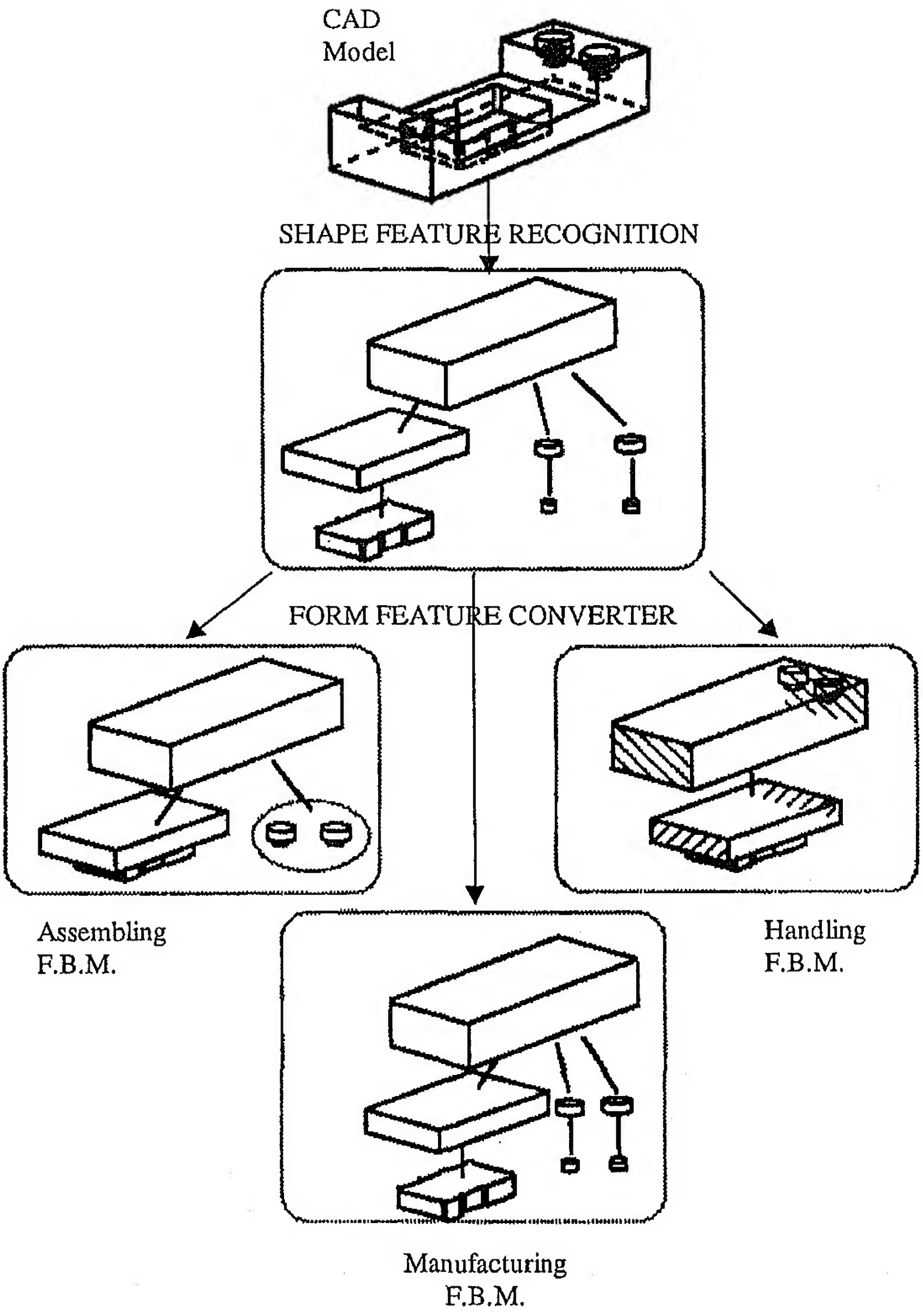


Fig. 2.2: Different context dependent FBM of the same object

Now, according to the context, it will be transformed to different feature based models. For example, manufacturing feature-based model will have primitives that correspond to a particular manufacturing method or process for creating that feature. For assembling context, only the compound feature formed by the couple of multi-diameter holes obtained joining the couples of holes, which are in contact with each other and the slot, are significant. On the contrary, for the handling purposes, the interesting features are the pairs of parallel faces in the stock material and in the slot components.

2.2 NEED OF USER-DEFINED FEATURES

If a powerful and convenient capability for user-defined features can be provided, then the library of design-with features can be smaller and the need for combinatorial power is also reduced. However it is not going to be easy to make provisions for user-defined primitives. Each new feature-form will have a great deal of data and procedural information associated with it. It must, for example, have data about what other feature-forms can attach to it, what the resulting combinations create, and it and its combinations are to be evaluated for manufacturability and other life-cycle evaluation programs. It will be time consuming process for users to provide all this information, and in some cases it will be beyond the sophistication of the user to do so. A much more likely scenario is that systems can be customized by software vendors. That is, a basic set of features is common to all, and provisions are made which allow vendors to add (both before and after delivery) the special primitives needed for individual customers. In either case research upon which customized feature-forms can be based has not yet been done. Of course, it can't be done, at least completely, until we know more formally what a feature is.

2.3 DESIGN LIBRARIES FOR CAD

Qamhiyah [6] has presented a strategy for the automatic generation of customized libraries of form features and basic design shapes for CAD systems. Generation of the initial libraries of form features and basic design shapes involves the application of a recursive procedure on each model stored in the CAD system. The procedure alternates between a feature extraction stage and a coding stage where the extracted features and the reconstructed objects are coded. The system user assigns a threshold to determine which features to include in the feature library. Similarly, the user-specified threshold and the frequency of encountering a given basic shape in the set of CAD models stored in the system will determine which basic shape are included in the basic shape library.

When new objects are modeled using the CAD system, new feature codes and basic shape codes are generated. As the frequency of occurrence of new feature and basic shape increases, the libraries will automatically be updated through the appending of new features and basic shapes.

2.4 LIMITATIONS BY THIS APPROACH

- Although many different methods are proposed to recognize the features from an object description, only regular shaped objects can be handled. Most of the features discussed are holes, slots, pockets, etc for general mechanical components.
- All the feature extraction techniques are applied to the available CAD model of the component.
- This technology is not useful to the component made up of sculptured surfaces.

Chapter 3

REVERSE ENGINEERING

3.1 INTRODUCTION

Designers and manufacturers have always evaluated their own and their competitors' products before launching new ideas. Today, that process is highly systemized in a separate technology called reverse engineering. Kanchana J. [7] formally defines reverse engineering, as the systematic evaluation of a product for the purpose of replication. It may involve making an exact copy or incorporating improvements.

Other concepts related to reverse engineering are benchmarking, re-engineering and pre-engineering. Benchmarking measures products and services against those that are best in their class. Re-engineering in the context of product development is the same as redesign. Pre-engineering is deciding whether changing a product is practical and if so, how to change.

Although the reverse engineering process may seem to be the opposite of the conventional manufacturing process, in truth the overall concept of the two are quite similar [Fig. 3.1]. The main difference is that the existing prototype in reverse engineering embodies the product specification in conventional manufacturing. A second difference is that the generation of the CAD model from a design concept is an explicit process, while generation of CAD model from digitized data is an inferential process. But in the product life cycle, reverse engineering has multiple uses within the development phase; thus reverse engineering can be looked as a part of overall product development cycle [Fig. 3.2].

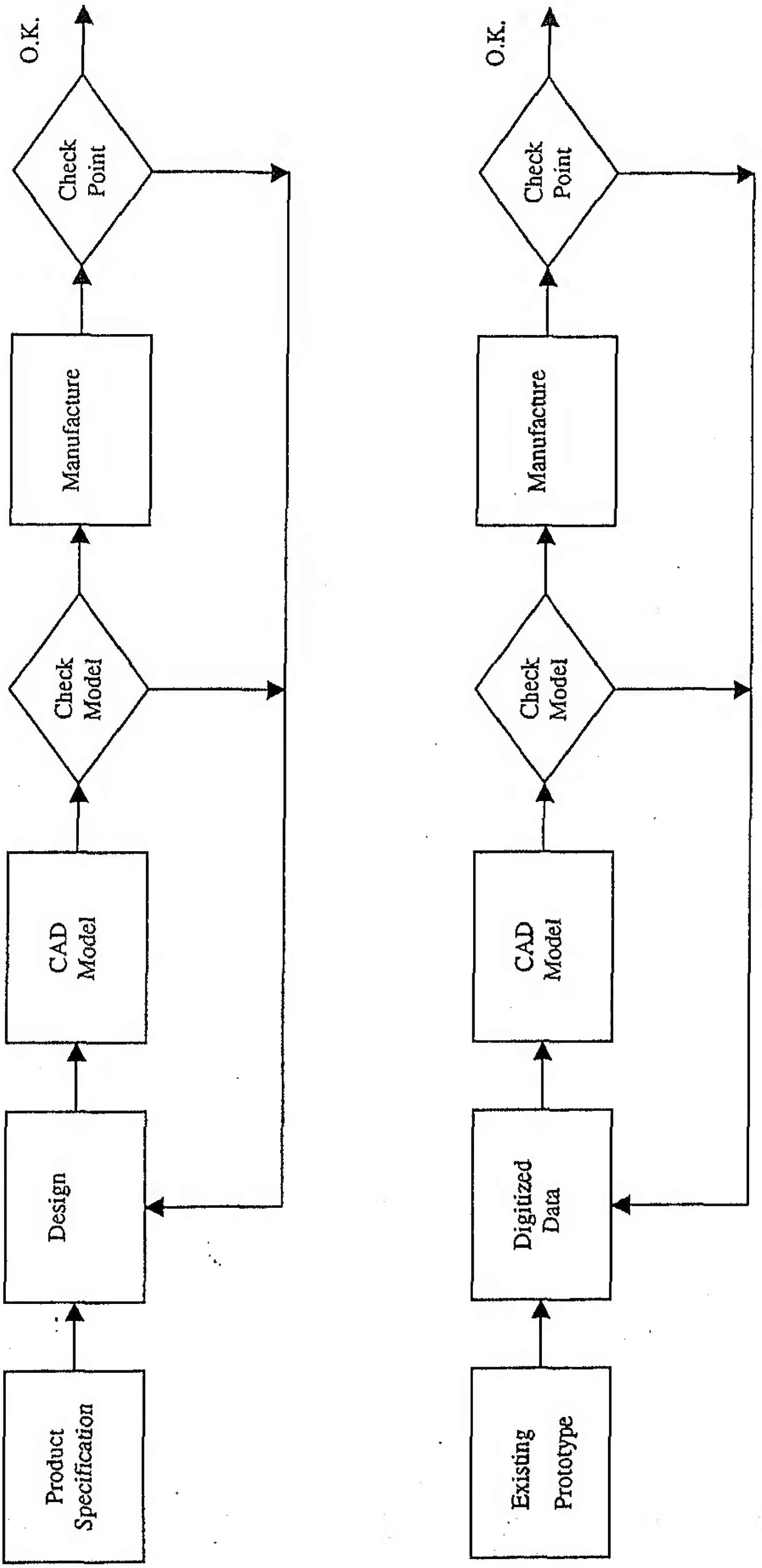


Fig.3.1: Conventional Vs Reverse Engineering

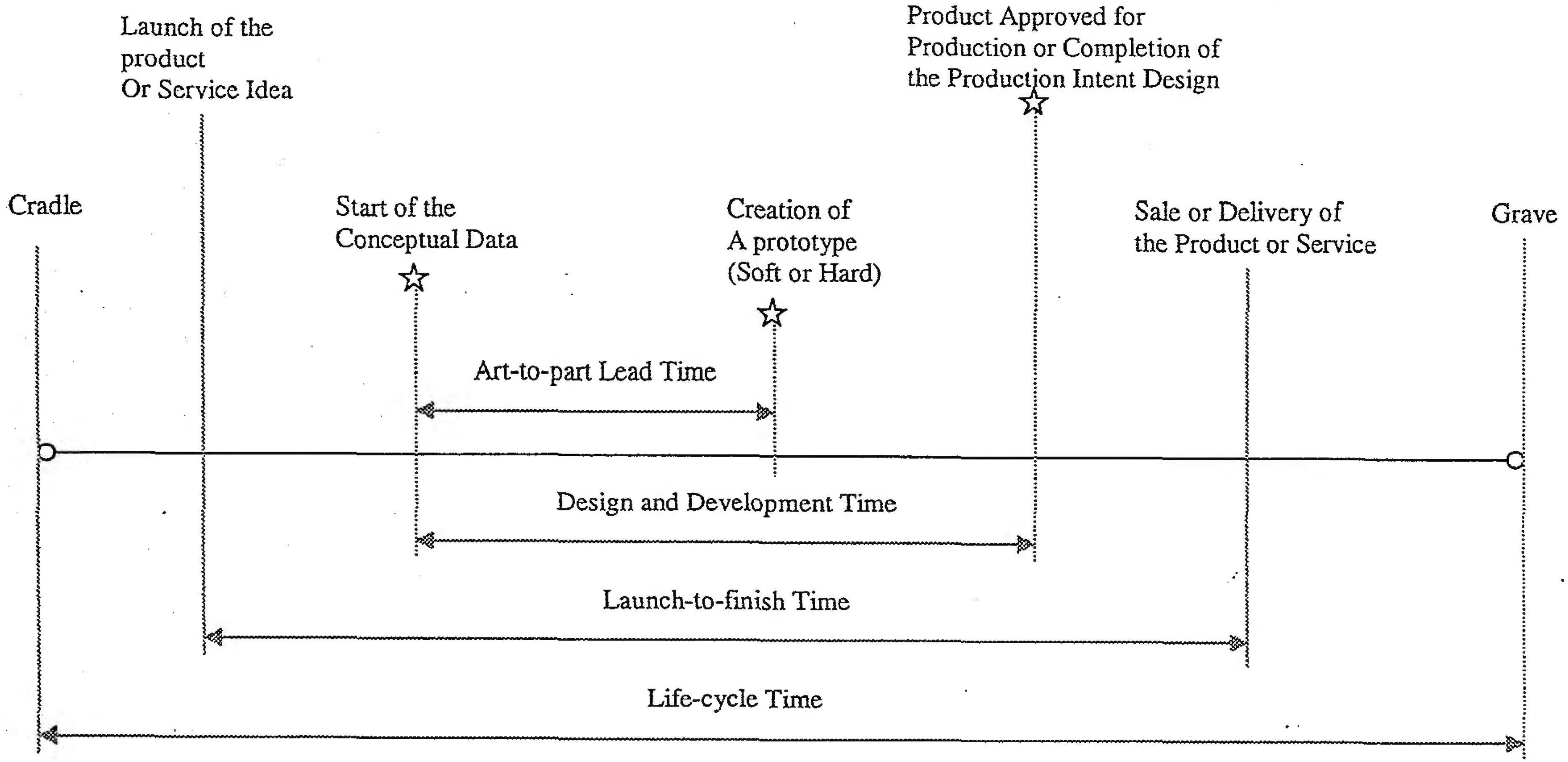

Launch of the product Or Service Idea
Product Approved for Production or Completion of the Production Intent Design
Cradle
Start of the Conceptual Data
Creation of A prototype (Soft or Hard)
Sale or Delivery of the Product or Service
Grave
Art-to-part Lead Time
Design and Development Time
Launch-to-finish Time
Life-cycle Time

Fig 3.2: Product Life Cycle depicting the places where reverse engineering play its role. [The places are shown by (☆)]

3.2 DESIGN FLOW OF REVERSE ENGINEERING

The procedure to construct geometric model i.e. continuous representation of a component from its discrete representation can be characterized by the flow chart [Fig. 3.3]

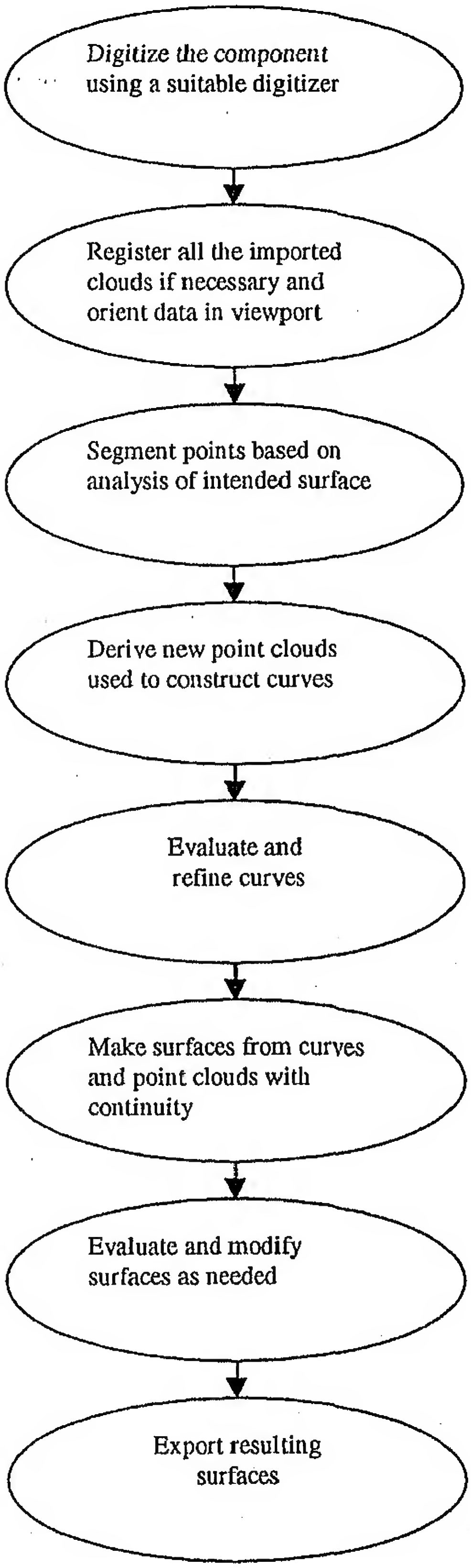
The main steps involved are described in next sections. The steps are illustrated with the process of making the surface model of a turbine blade. The component was scanned in two different positions.

3.2.1 Data Capture

Analyze the component for needed accuracy of surface fitting and for scanning. Check the accessibility of different features. Decide the type of digitizer required accordingly. Bernard A. [8] presents a state of the art of reverse engineering for rapid product development, wherein he describes various sensors for digitizing a part.

The digitizers/sensors are divided into three families according to the acquisition mode. Two main families gather the sensors using a passive acquisition technology and those using an active technology. The active mode sensors are divided into two main families: contact sensors and non-contact sensors. Each family can still be divided into subfamilies. Thus, the contact sensors are subdivided in two subsets: point-to-point and analog sensors. In the same way, the non-contact sensors can be subdivided in several subfamilies: the optical sensors not using a laser, those using one, the sensors used more usually in medical imagery and those using the principal of the flight time. The third family comprises only CGI sensor ("destructive combined mode").

Scan the component by a suitable digitizer in one or more sets by either orienting the component or by moving the scanner to different position, according to the size and topology of the component. The two point data sets for the turbine blade before post-processing [Fig. 3.4] is shown.



Digitize the component using a suitable digitizer
Register all the imported clouds if necessary and orient data in viewport
Segment points based on analysis of intended surface
Derive new point clouds used to construct curves
Evaluate and refine curves
Make surfaces from curves and point clouds with continuity
Evaluate and modify surfaces as needed
Export resulting surfaces

Fig. 3.3: Reverse engineering design flow

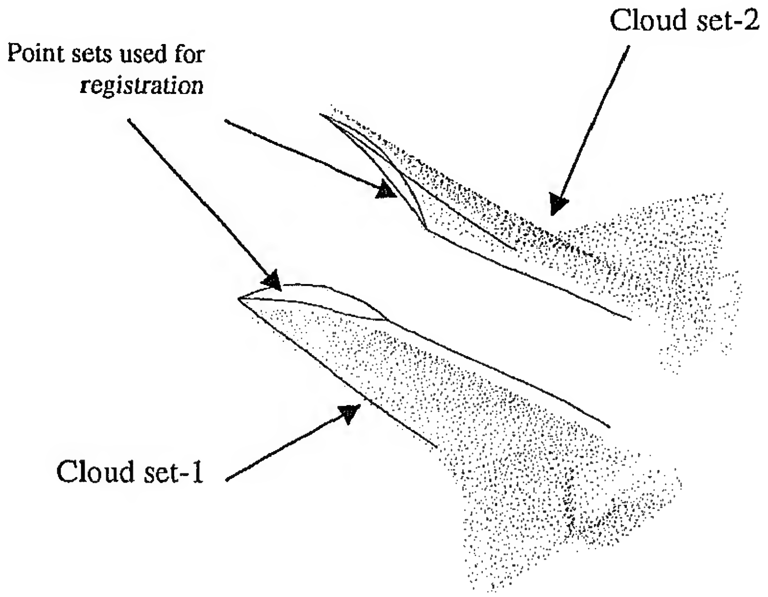


Fig. 3.4: Point cloud prior to post-processing

3.2.2 Post-processing of the Data

Import scanned data points into the system. If the component is scanned in more than one set, registration is required. Registration is the process of bringing geometric entities into proper alignment. It is necessary when an object or parts of an object have to be scanned at different times, or have to match coordinate systems of other scanned data. One may need to register data when an existing model must be registered with scanned data. The entities, which aid in registration, are sets of points or curves or surfaces that are present in all the point sets. In registration either these entities are aligned or in some cases, curves or surfaces are aligned with the corresponding point clouds in other point sets.

Orient imported data in viewport for segmentation. Scattered points that are introduced due to error in sensors or by manual operation are removed. Segment points based on analysis of intended surface. To do the segmentation, point cloud is triangulated which resembles to a skin wrapped around it. It is analyzed for curvature variation. With the help of this variation, point cloud is segmented into different parts.

Sampling of the point cloud may be needed to get uniform density of the cloud. In this process, points within the specified distance tolerance are removed.

If scanning is manual, it is advisable to segment the point cloud during the scanning process, according to the different features of the component. When the component is made up of free form surfaces, during post-processing, segmentation may be done with the help of curvature features of the point cloud. Chordal based or angular based features can be extracted from the point cloud. Point cloud of turbine blade after post-processing is shown [Fig. 3.5].

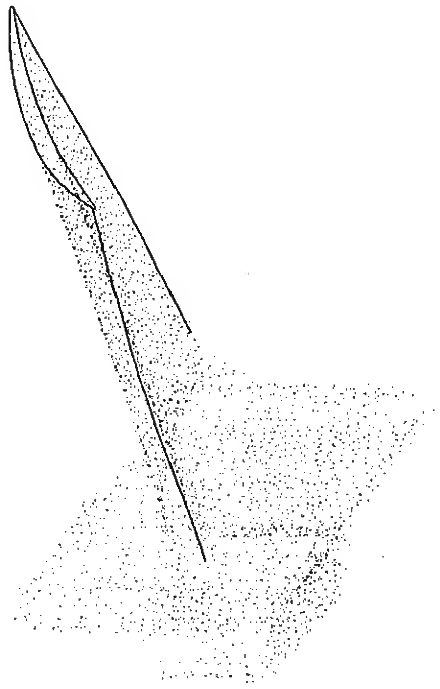


Fig. 3.5: Complete point cloud after post-processing which includes registration

3.2.3 Curve Fitting Process

Curves are fitted into the point clouds, according to the intended surfaces. [Fig. 3.6] Standard forms, like straight lines, arcs and circles can be fitted for prismatic surfaces. For free form surfaces, point cloud is cross-sectioned into a number of thin segments. Free form curves are fitted into these scan lines. The number of control points for the curves are determined on the basis of accuracy and intricacy of the required surfaces. For more accuracy and for high curvature regions, number of control points is more and vice versa. Smoothness of the curves carves is reduced with increase in number of control points. When curves are used for cresting a lofted surface, they are to be cleaned and reparameterized. During cleaning, knots are removed from a curve so that its parameterization is optimal for the given shape. During reparameterization, to manipulate the number of control points or spacing between the knot points.

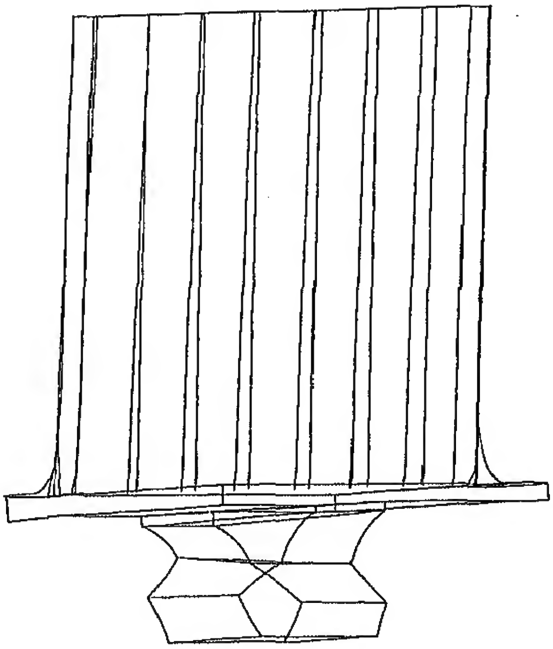


Fig. 3.6: Curve network for the turbine blade

3.2.4 Surface Fitting Process

Fit prismatic surfaces, like planes, cylinders, spheres or cones into the corresponding point segments. Free form surfaces can be created using the curve network. Or it can be created either by the use of UV curve network, by sweeping the generator curve along one or two path curves. Curves fitted to the distinct edges of the point cloud create the boundaries of these surfaces. Continuity can be maintained while creating these surfaces. To fill the gap between different surfaces created by the use of above methods, surfaces are created by blending with boundary curves or by creating the fillet or merge surfaces [Fig. 3.7]. After completing the patch network, different surfaces are stitched to create a watertight geometry and the resulting model is exported to different CAD system for further operations.

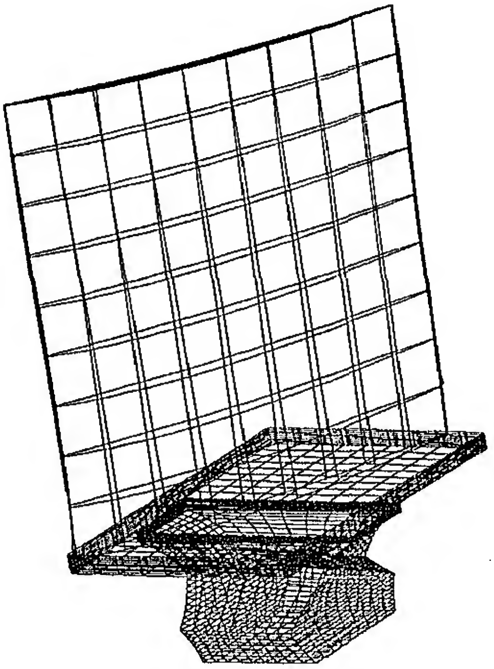


Fig. 3.7: The complete surface model of turbine blade

3.3 APPLICATIONS OF REVERSE ENGINEERING

- To generate the CAD model of a newly designed component made up of free-form surfaces, by scanning a wood or clay model.

3.3 APPLICATIONS OF REVERSE ENGINEERING

- To generate the CAD model of a newly designed component made up of free-form surfaces, by scanning a wood or clay model.
- In animation or rendering applications, where the complexity of the objects is much higher and the need for precision is often reduced. Highly irregular, free form sculptured objects like people, animals, etc are to be rapidly represented.
- To modify the existing design of a part, by making the shape changes in its component surfaces. The changes can be applied to either CAD model directly or to the existing part, which is then digitized to create the CAD model.
- To produce rapid prototype of a component directly by triangulation of digitized data. This greatly reduces the production cycle time.
- In CAD to part inspection or part to part inspection. In CAD to part inspection, part is scanned and digitized data of component is analyzed with respect to the surfaces of already existing CAD model. Whereas, in part to part inspection, both the components are digitized, and the difference between both the data sets is analyzed.
- Chivate P. and Jablokow [9] quote the usefulness of generating solid-model from measured point data for the inspection of manufactured components and assemblies, assembly fit-up analysis and uncertainty analysis.
- To reduce data greatly in computer graphics, cartography and other applications that leads to more efficient visualization algorithms and a large decrease in storage requirements. In many cases, data reduction is crucial if they have to be transmitted.

Two case studies of solid model generation through reverse engineering are discussed in Appendix. Faro Arm was used to scan the models in multiple set-ups. Using Imageware's Surfacer (Version 8.0), surface models were generated. In both the cases, probes used were spherical probe of diameter 0.125" to scan the entire surfaces and point probe to scan boundary points. Post-processing on the surface models was done in SDRC's I-Deas (MS 6) software to get the final solid model.

Chapter 4

EXTRACTION OF CURVE BASED FEATURES

4.1 INTRODUCTION

This methodology of feature extraction can be applied to digitized data of any component. Surfacer V 8.1, a commercial software [10], has been used for this purpose. The sensor used for this work is Faro Arm, a portable multiple axis articulated arm with single point repeatability two sigma equal to +/- 0.003". Digitized data is imported into the viewport [Fig. 4.1 (A)]. Registration of all the point clouds is done, if required, to get a single point cloud data.

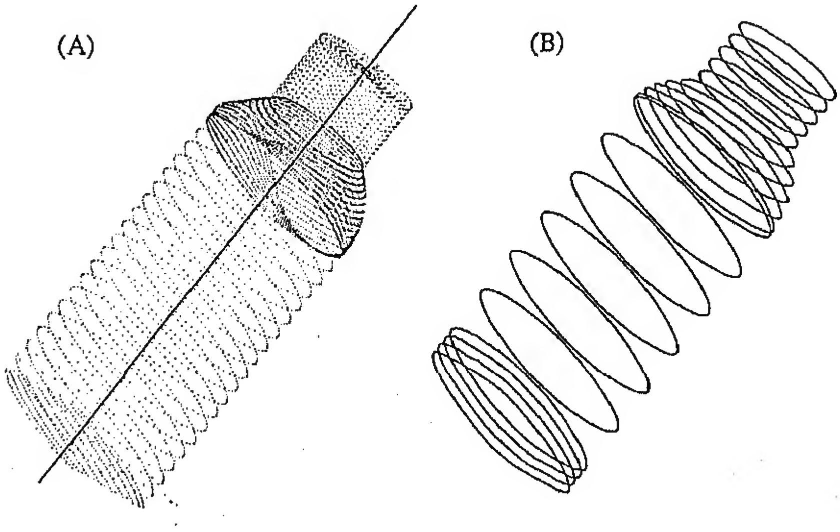


Fig 4.1: (A) Complete point cloud with primary direction,
(B) Parallel cross-sections created on original cloud

4.2 POST-PROCESSING OF POINT CLOUD

Post-processing tools, as described in section 3.2.2 are applied to the point cloud. It is then analyzed for primary direction [Fig. 4.1 (A)]. Primary direction can be formally defined as the direction along which, the spread of points is maximum, for example, axial direction in case of point cloud of a cylinder. Cross-sections are made on the point cloud along the primary direction [Fig. 4.1 (B)]. Here, number of cross-sections to be made depend upon the complexity of the part. If the part is made up of a many surfaces, more cross-sections are needed and vice versa. For making a feature model, the number of cross-sections and their locations are decided manually as described in section 3.2.2.

4.3 GENERATING THE GRAPH FOR FEATURE EXTRACTION

An interpolating closed B-spline curve of degree three is fitted into a cross-section of points. The number of control points is the same as that of number of points in that cross section. Between approximating and an interpolating curve, the obvious choice for geometric modeling is an approximating curve, which is smoother. Also various curve segments can be smoothly blended using approximation. Also in most of the downstream applications, such as NC machining, discontinuities are not desirable in tool path. When a designer wants to fit an approximating curve through a point data set, he/she has to specify the number of control points for that curve. If tight tolerances are desired, higher number of control points are to be specified and for loose tolerances, required number of control points is lower [Fig. 4.2]. The knot points are also shown for an approximating B-spline [Fig. 4.3]. But the number of control points has an adverse effect on smoothness of a curve. If the designer is not experienced, it becomes difficult for him/her to exactly specify the number of control points to approximately fit a B-spline through a set of data points and it has to be evolved by trial and error method. That is why, interpolating curve is chosen for the feature extraction purpose, which will also be useful if one wants to automate the process.

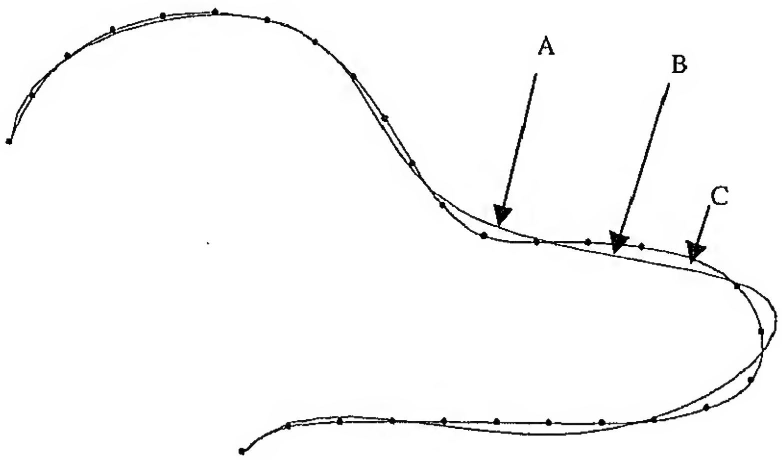


Fig. 4.2: Curve: (A) Approximating B-spline with n=8, (B) Approximating B-spline with n=16, where, n = number of control points (C) Interpolating B-spline.

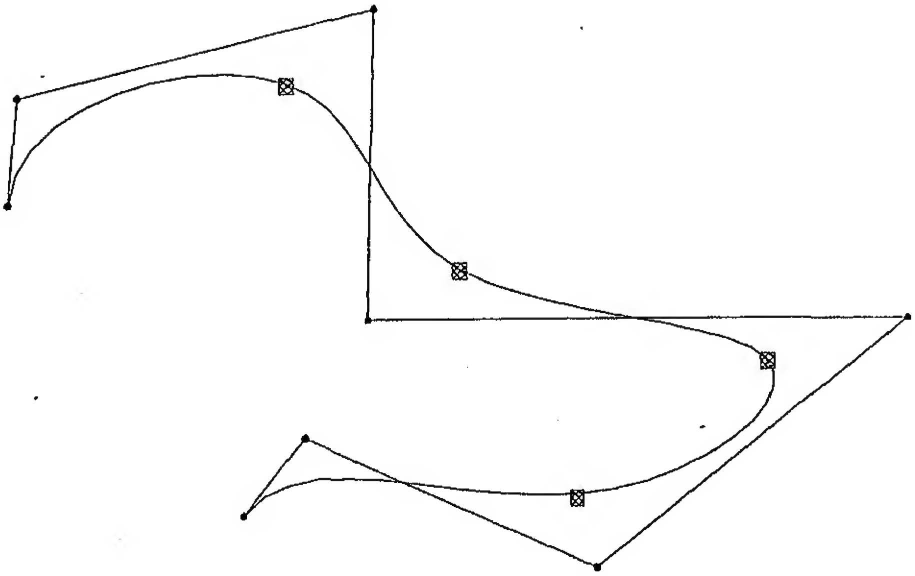


Fig. 4.3: Knot points (▨) and control points (•) for an approximating B-spline with 8 number of control points.

When an interpolating curve is fitted into a point data set, it is not smooth and have large fluctuations in its curvature. These fluctuations make it difficult to differentiate parts of this curve into feature curves. A curve with degree one is nothing but a set of straight lines passing through each and every point. A curve with degree two does not have curvature continuities between its various segments. Considering these points, the degree of curve is kept three.

On a planer curve, there are basically two classes of feature points, namely corners and smooth joints. These feature points are used to identify sharp corners with tangent discontinuities and smooth joints with curvature discontinuities respectively. Corners and smooth joints partition a planer curve into a number of primitive curve segments or curve features. In the context of curve construction, these curve features can be categorized into three classes, namely primary, secondary and auxiliary curve primitives. Primary curve primitives are the most important curve segments that form the major skeleton of a planer curve, such as long line segments, skeleton arcs, or main free form curve segments. Secondary curve primitives are mainly those curve segments for connecting the primary feature primitives, such as chamfering line segments, filleting arcs, or small free-form segments for bridging two primary curve features. Auxiliary curve primitives refer to those local minor curve features as a result of auxiliary or minor geometric constructs of the object. Such minor geometric constructs cause the planer curve to locally deviate from its ideal shape. For a complex contour, the subdivision of curve primitives is often not unique.

A new point set of control polygon vertices is derived. The graph with the number of control polygon vertices on X- axis and the angle between three consecutive vertices on Y-axis is generated.

4.4 EXTRACTION OF CURVE BASED FEATURES

When the required graph is generated, it is analyzed to extract the desired feature curve segments. This is done on the basis of variation in angle between consecutive control plot vertices of the interpolated B-spline [Fig. 4.3].

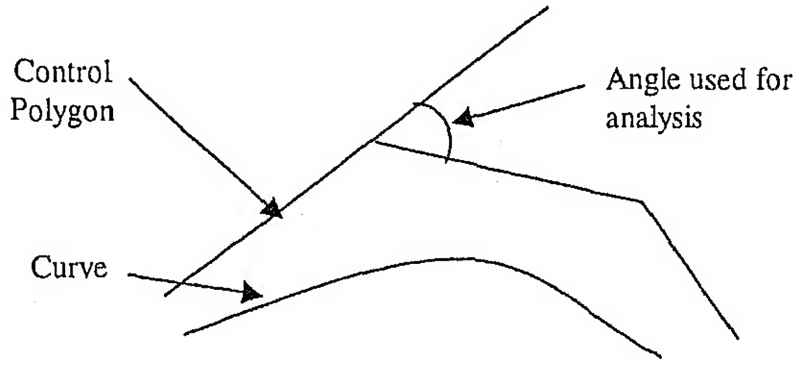


Fig 4.4: Angle used for extraction of curve based features

Every curve feature, mentioned above, shows peculiarity in its control polygon shape [Fig 4.5].

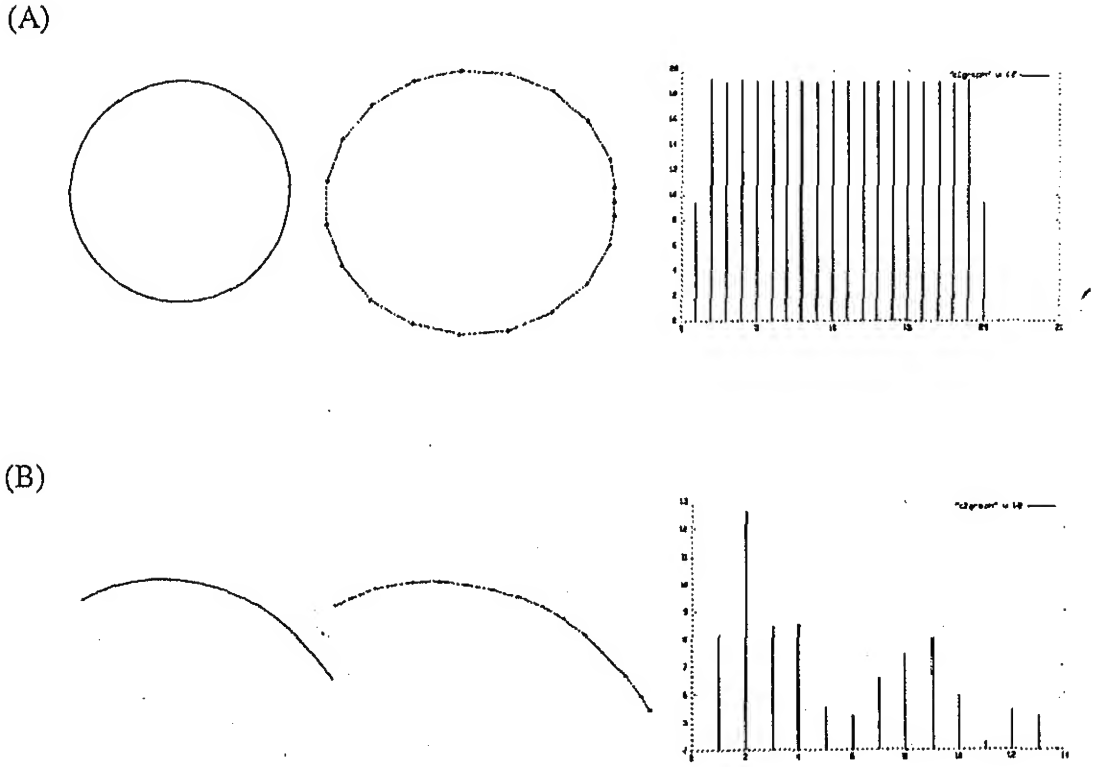


Fig 4.5: (A) Circle, (B) Spline segment

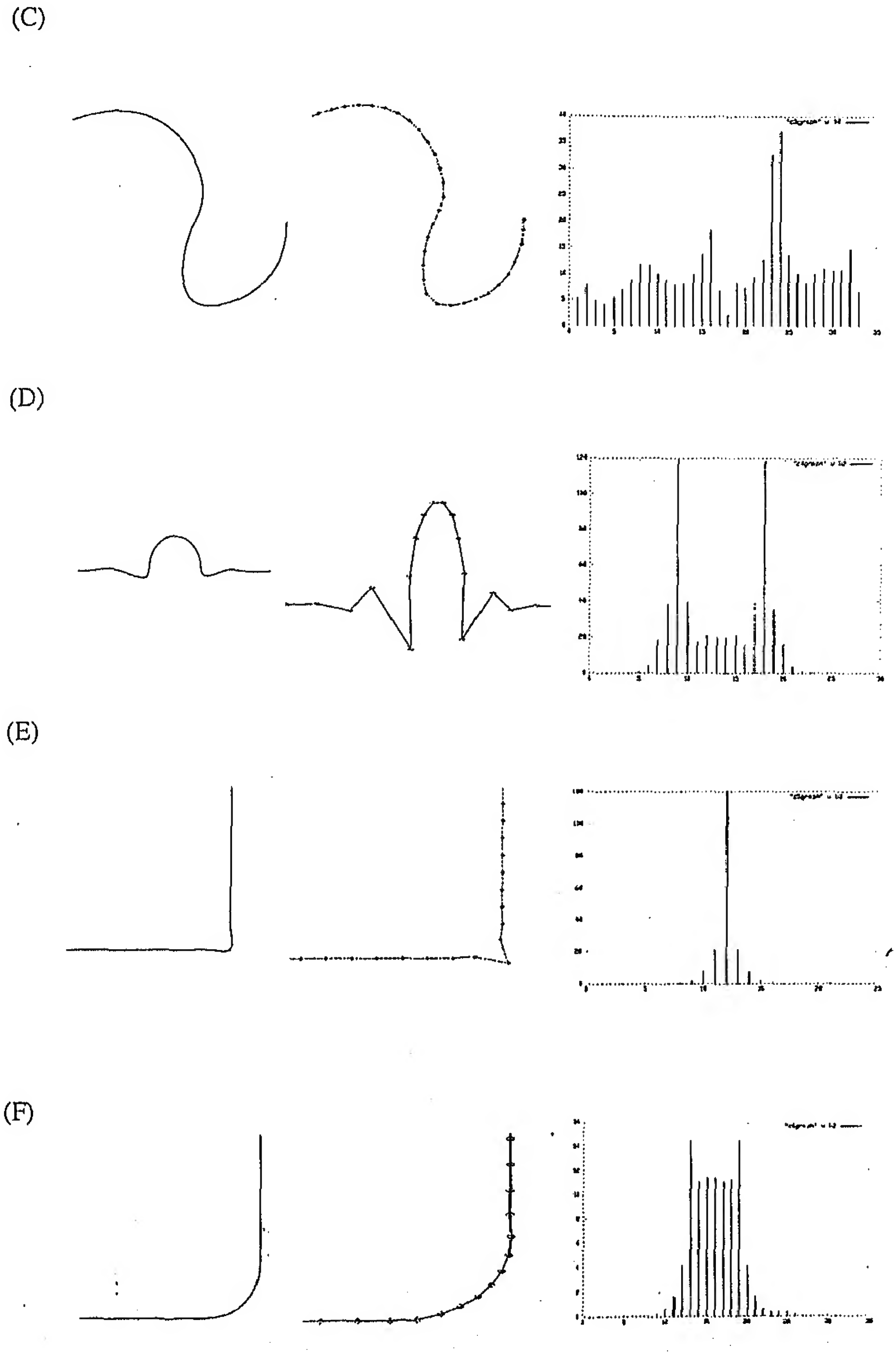


Fig 4.5: (C) Spline with two segments, (D) Auxiliary curve, (E) Curve with sharp corner, (F) Two splines joined with fillet curve.

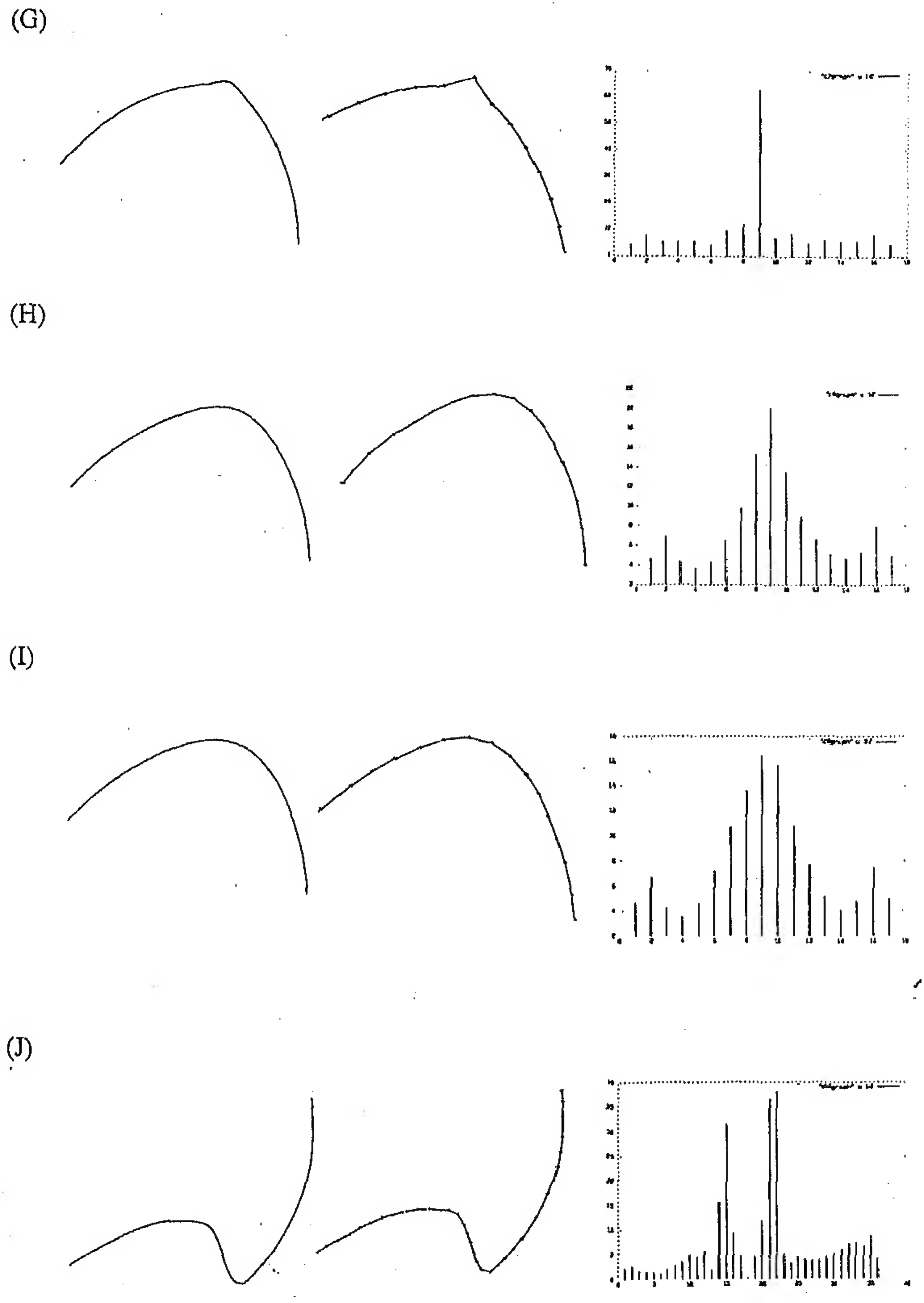


Fig. 4.5: Two splines joined with, (G) Position continuity, (H) Tangent continuity, (I) Curvature continuity and (J) Merge curve

In the polygon of primary feature, the desired angle does not change drastically and the variation is constant. For the secondary feature, this angle varies from 15 to 45 degrees. When two different features merge into each other, this angle varies from 0 to 2 degrees. Thus changeover from one feature to another can be tracked from the graph.

It is very important to analyze the variation of this angle for the extracted feature. For primary feature this value may remain constant for some portion of curve or it may fluctuate at some places from 5-15 to 0 degrees. It is due to occurrence of inflection points. It may be noted that secondary feature is either a merge curve or a fillet. But the relative position of control points differentiates between these two types. For a fillet, the position of control point is symmetric, which may not be the case of a merge curve. For auxiliary feature, variation of this angle is similar as in case of secondary feature. At the merging of auxiliary feature, there is sudden change in angle due to position continuity wherein at the merging of secondary feature, this angle is less than 2 degrees due to tangent or curvature continuity.

4.5 AUTOMATING THE PROCESS OF FEATURE EXTRACTION

One can recognize the desired features directly from the graph mentioned above, which is manual process. To automate this process of feature extraction, pattern recognition tools can be applied to this graph. Other way is to make templates for all the features, which then can be applied to each point data cross-section to extract the desired features automatically. To get the idea of this automatic process, some sample codes in C language have been developed. The next steps to be followed are described below.

After getting the cross-section out of the point cloud, interpolating B-spline is fitted into it. A new point set is derived from the control polygon vertices of this curve. A code to extract the points and to compute the angle between three consecutive points is applied to it and the results are saved.

This file is used to generate the desired graph. The variation in this angle can be separated into different ranges, which correspond to particular feature and feature-to-feature merging. First the output file is traced for feature-to-feature merging. When two consecutive merging places are found, the angle values between these two values are analyzed to check that to which feature it correspond. The checking is done, using the mathematical conditions in the "if-else" form. The output of this code consists of the positions of control polygon vertices where the merging of different features occur and the types of feature in that cross-section.

4.6 FEATURE DEFINITION FROM THE POINT CLOUD

Once all the features, present in a particular cross-section are extracted, the optimum number of knot points for each of them has to be obtained, which is done as follows. For a B-spline,

Number of knot points = Number of control point - n

Where,

n=4, for open B-spline.

n=3, for closed B-spline

Knot points are the points on an approximating B-spline, which divide it into different segments [Fig 4.8]. For a primary feature, the knot points can be identified as the points near to the control point of that curve which represent a crest in the graph. So the exact number of knot points are equal to the number of crests on the corresponding graph plus two (to include endpoints).

For fillet, the position of knot points is symmetric whereas in case of merge curve, it is asymmetric. It is noted that the extent of secondary and auxiliary features distinguishes them from primary features.

Once we have the required data for fitting an approximating B-spline into the given cross-section, it will be used for extracting the surface features from the whole point cloud.

Chapter 5

EXTRACTION OF SURFACE BASED FEATURES

5.1 INTRODUCTION

In geometric modeling, one can classify the surfaces into major categories as, surfaces of revolution, sweep surfaces, quadratic surfaces, ruled surfaces, coons surfaces, bezier surfaces and B-spline surfaces. A single surface can be modeled by the use of different techniques. In this work, the curves, which are parallel to each other, are generated as an initial step towards surface generation. The technique of lofting is used to model the surface features, which is a general procedure in reverse engineering. Later on, when all the surface features are extracted and the feature model of the given object is created, precise parameterization of each surface feature is done according to the connectivity of these features with each other.

Depending upon the type of curve features, the surfaces can be divided into three different categories. Using the philosophy of method of sweep surface creation does these categories. When a straight line is swept along a straight line, a planer surface is generated. When a straight line is swept along a 3D-curve i.e. nonlinear space curve, a ruled or singly curved surface is obtained and after sweeping a 3D curve along a 3d curve, a doubly curved surface is obtained. Thus to define a swept surface, the curve which is swept i.e. generator and the curve, along which the generator sweeps i.e. path curve, need to be defined.

5.2 EXTRACTION OF SURFACE BASED FEATURES

Applying the curve based extraction techniques to every cross-section containing point data, we obtain the feature points for each curve feature [Fig. 5.1].

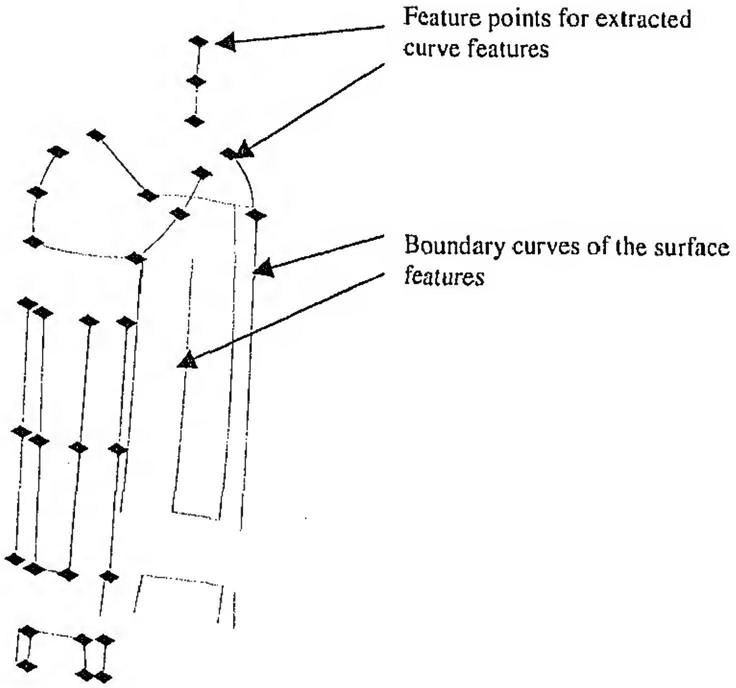

Feature points for extracted curve features
Boundary curves of the surface features

Fig 5.1: Network of feature points and boundaries of surface features

In a 3-D space, a collection of feature points among consecutive layer form the edges or boundary curves of the surface features to be identified. The first class of feature points, i.e. the corners of planer curves, form the first class of edges, namely crease edge. They are also called fold or roof edges of 3-D surfaces. The second class of feature points, i.e. the smooth joints, form the second class of edges, namely smooth edges of 3-D surfaces. Similarly, a collection of feature primitives, such as the primary, secondary and auxiliary curve primitives of the planer contours, also form feature primitives in 3-D space, namely primary, secondary and auxiliary surface primitives, respectively. The classification of a 3-D surface into surface primitives is usually not unique either. Segmentation of a 3-D surface into user-interested surface primitives or feature contours is described in next paragraphs.

As described in previous chapter, the curve features are obtained on each point cloud cross-section. These curves are grouped together so that after lofting the curves in a group, a single surface feature is obtained in 3-D space. The grouping can be done as described below.

The process starts from the first cross-section of point data. We obtain different types of curve features on each cross section. A counter is assigned to the number of feature curves on each cross-section. Proceeding to the next cross section, if this counter matches with that of the previous one, it shows that no new surface feature is countered. If the two counters don't match, it indicates the presence of a new surface feature. Another possibility is that even if the counter matches, the type of curve features may be totally different. In both the cases, we group the curves, which are obtained as the collection of feature points such that the type of curve features are same for each cross-section of point cloud. Post-processing is necessary on the extracted surface features [Fig. 5.2]. It consists of extending the surface patches until they intersect with each other. Also these surfaces need to be trimmed to get the final CAD model [Fig. 5.3].

Similar to curve features, surface features can also be divided into three categories, which are primary surfaces, secondary surfaces and auxiliary surfaces. Primary surfaces are the surfaces, which contribute maximum to the surface area of a surface model. Secondary surfaces are mainly those patches that connect the primary surfaces, such as fillet surfaces and merge surfaces. Auxiliary surfaces are those which contribute minimum towards the surface area of a surface model. These are generally tangent discontinuous with the neighboring surfaces. Sometimes we may not get each and every boundary point on the surface features. These are obtained by extending the boundary curves for the surface features. Applying some rules does merging of the surface features along primary direction. When the nature of curve features changes while going from one cross-section to the next one, it indicates the presence of secondary feature along the primary direction. Similarly, when the number of curve features changes while going from one cross-section to the next, it indicates that the features are merged into each other with positional continuity.

A primary surface feature may consist of inflection points. So it can be divided into different surfaces at these points. As described above, the surfaces can be categorized into either planer, singly curved or doubly curved surfaces. So one can specify generator curve and a path curve to specify the type of surface or if the shape is

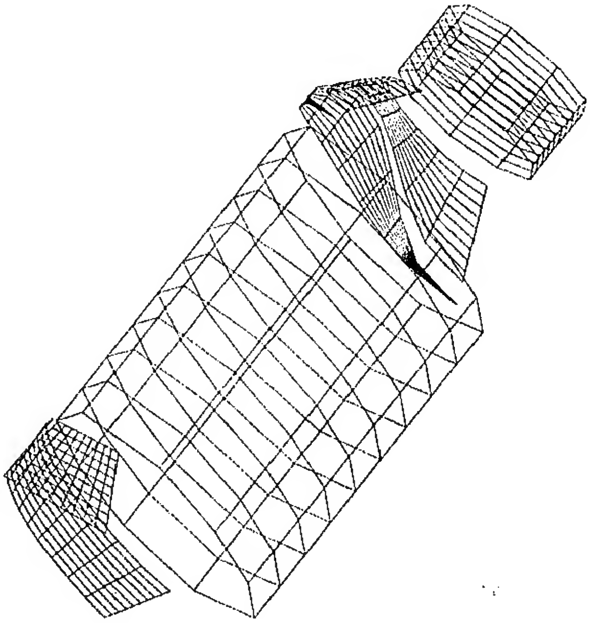


Fig 5.2: Extracted surface features

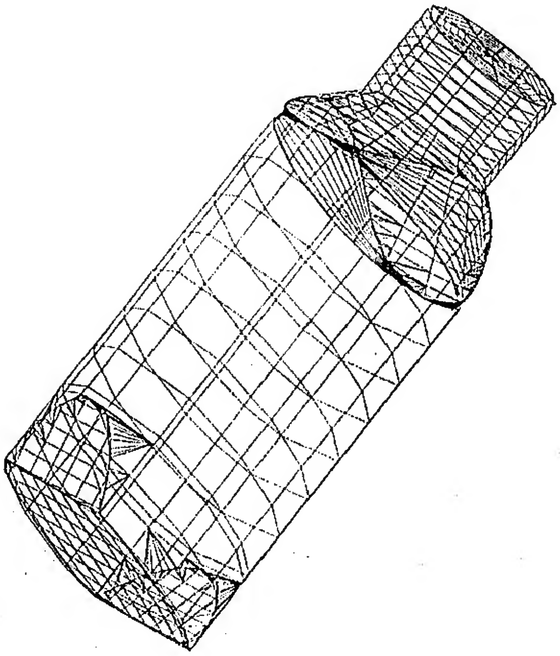


Fig 5.3: Complete surface model

too complex, one needs to specify the curves at different locations as in case of loft surfaces.

A primary surface feature may consist of inflection points. So it can be divided into different surfaces at these points. As described above, the surfaces can be categorized into either planer, singly curved or doubly curved surfaces. So one can specify generator curve and a path curve to specify the type of surface or if the shape is too complex, one needs to specify the curves at different locations as in case of loft surfaces.

5.3 DEVELOPMENT OF FEATURE MODEL AND INCORPORATING PARAMETERIZATION

Once we obtain all the surface features, it is necessary to group them together to define the features that will preserve the design intent by the designer. The process of grouping is manual. Although there are no specific rules for grouping the surface features, some logical ways can be followed. For example, all the surface features that are on the same height may be grouped to form a single feature. Also, if more than one surface features can be created by using a single technique of surface generation, these features can be grouped together. Thus the feature model for the given point data set will consists of the features made up of surface features which in turn will consist of one or more patches.

After creating the feature model, it is analyzed to obtain the datum points in the context of parameterization [Fig. 5.4]. These datum points are different than the feature points, discussed in the previous sections. These are the points that lie on the datum axis and are useful to locate the position of the planes π_I, parallel to the datum plane. The position of these planes is represented by the height vectors h_i and contains the curve vertices in fourth layer of the feature model. The location of curve vertices is referenced with respect to the datum points on the corresponding plane.

To define the parameters, we define a datum axis, which runs along the primary direction and passes through geometric center. The geometric center can be obtained by averaging the co-ordinates of all the points in the cloud of the object. So it is necessary to have the density of the point data set to be uniform. It is achieved by having the point data set dense and applying space sampling techniques to it.

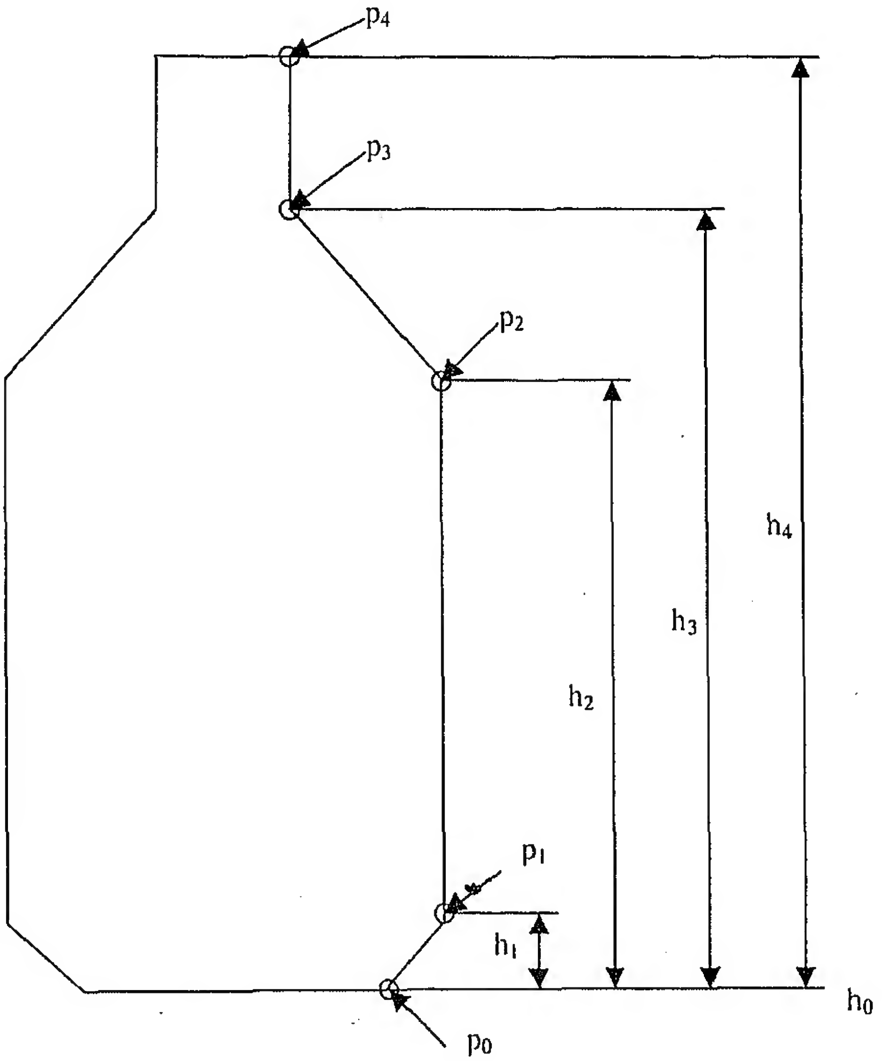


Fig 5.4: Feature model showing the parameters and reference planes

If the object is symmetrical, the axis of symmetry can be used as the datum axis. If the object is unsymmetrical, the datum axis is located manually, which will run along the primary direction. Also we define the datum plane at the bottom of the model. The datum plane is perpendicular to the datum axis and all the measurements for the purpose of parameterization will be done from this datum plane and along the datum axis.

A feature in the feature model is defined by a set of surfaces. A surface is defined by two curves and according to the philosophy of sweep surfaces, as described above, one of them is a generator and the other, a path curve. Thus the feature model of a given object can be represented by a four-layer structure network [Fig. 5.5]. The first layer is the feature layer consisting of a graph of the constituent features and their connectivity. A two-headed arrow represents the connectivity of features. The second layer is a surface feature layer. It consists of all the constituent surfaces of the corresponding features in the feature layer. A single-headed arrow indicates the relation between a surface and a feature. A two-headed arrow represents the connectivity among the different surfaces. The secondary surface (S') i.e. fillet surfaces or merge surfaces, is also indicated on the connecting arrow of corresponding surfaces. The third layer is the curve feature layer. It consists of all the constituent curves of the corresponding surface features in surface feature layer. The information in this layer consists of the type of curve, the parameters to define that curve and its vertices. The parameters of the fillet features in the surface feature layer are also represented in this layer. These parameters are the radii of fillets. A fillet may be of uniform radius or of varying radii. If the secondary surface is a merge surface, the type of continuity i.e. C0, C1 or C2 between the corresponding surfaces is also represented. The last layer is the layer of vertices of the curve features. It contains the co-ordinates of each vertex. These co-ordinates are referred to the datum points on the corresponding plane. This plane is parallel to the datum plane.

5.4 INTEGRATION OF FEATURE BASED DESIGN AND REVERSE ENGINEERING

Once we create a feature model for an object, it can be used to extract the features directly from the point cloud of a similar object. The steps involved are as described below.

Point cloud of the similar object is aligned with the feature model. For this purpose, the pre-processing is done on the point cloud and the datum plane and datum axis is located. Using the datum plane and the datum axis, alignment is done. Cross-sections are created on the point cloud as in case of feature model. The point cloud is

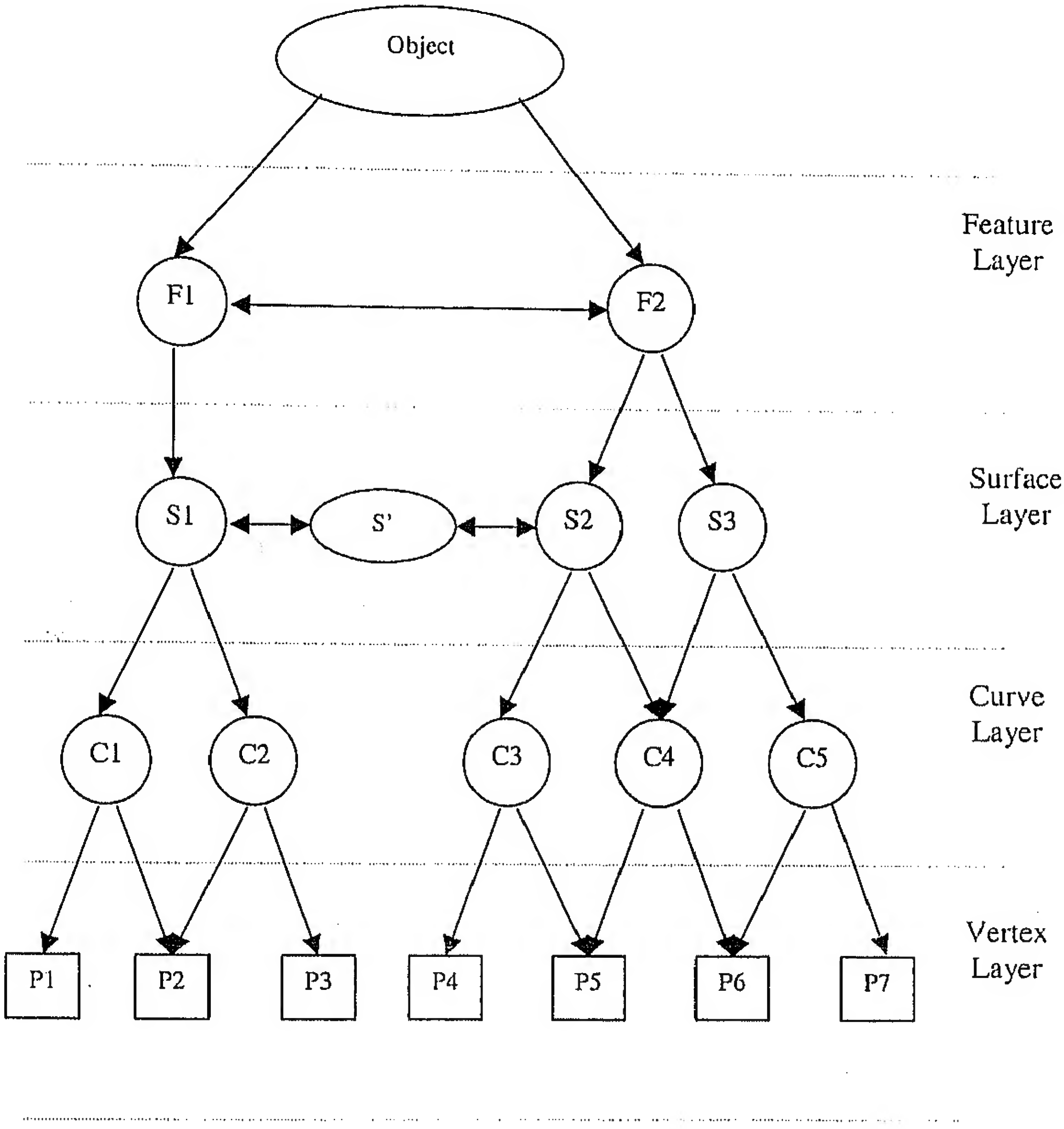


Fig. 5.5: A four-layer structure network of feature model

cut by the planes, which contain the feature points and curve features. A 3-D interpolating B-spline of degree one is fitted into the point cross-section so as to extract the feature points. These feature points are obtained in a way, similar to the extraction of curve features. These feature points locate the planes π_i for the point cloud. Also the curve features that are necessary to define the features are also extracted. Using these curves and the points, scaling of corresponding features on feature model is done and new surface patches are created. Post-processing which consists stitching and error analysis is done to get the complete watertight surface model of the new object.

5.5 DEVELOPMENT OF FEATURE LIBRARY

In industrial applications, surfaces are often classified as belonging to one of two categories:

1. Freeform surfaces
2. Functional surfaces

Freeform surfaces, also called as aesthetic or show surfaces, are characterized by their smooth curvature distributions, while functional surfaces are characterized by highly irregular, multifeatured shapes consisting of pockets, ribs and channels comprising a surface geometry of highly varying length scales. Examples of freeform surfaces are automobile outer skin panels, television boxes, etc. Examples of functional surfaces are interior surfaces of injection moulds, internal surfaces of IC engine cylinder blocks, etc. Out of these two categories, shapes of freeform surfaces can be modified without much concern about the tolerance values. This process has been standardized in terms of feature technology.

. In this feature based design technique, the basic unit is a feature and products are constructed by a sequence of feature attachment operations. The type and number of possible features involved depend upon product type, the application reasoning process and the level of abstraction. The usefulness of the mechanism relies on two functional capabilities. First, the shape and size of the user-defined features are instantiated according to parameter values given by the end-user. Second, the end-user positions and orients the feature in the part being designed by means of geometric gestures on geometric references. The methodology developed in this work is useful to obtain a variety of surface models, by using the features that are already developed. The strategy is, taking advantage of the known, geometric structure of the data being fit; a generic model having this shape is used as an initial approximation. After the feature model for an object is developed, an application specific feature library is developed. These features are assigned the parameters that will describe its geometry and topology. The features may consist more than one surface patches. The constituent curves of these patches are assigned the parameters. This is similar to the four-layer structure as described in section 5.3. Thus when a designer wants to create a surface model for any product, it can be done with the help of these features. The process is described in the next paragraph.

If a feature in the feature model is to be modified, the feature library is searched in the class of same feature type for different geometry and topologies. If any of the features is chosen, the parameters are modified to fit into the model in hand. The feature thus obtained will be added to the same type of feature class in the feature library. During this upgradation, an attempt is made to modify the definition of that feature class to make it more generalized.

5.6 CASE STUDY

The object is taken from consumer products, as it is the area where the aesthetics of the product need to be appealing to consumers. Point data set consists of 20000 points. As the object is symmetric about a plane, only one side of it is scanned and is mirrored about that plane. 0.25 inch ball probe is used to scan the object and point probe is used to scan the reference points for obtaining mirror plane location. In preprocessing stage, a new cloud is obtained by offsetting the original cloud by the radius of the ball probe. The cloud is sampled using space-sampling technique with sampling distance of 0.25mm. The number of points is reduced to 14000. The point set is mirrored about the mirror plane to get the total cloud. The axis is manually extracted along which the cross-sections are created on the point cloud. The cross-sections are made interactively so that there are more cross-sections where there is a probability of having complex feature.

As the data is noisy, cross-sections were filtered using Gaussian filter. To extract the features, interpolating 3-D B-spline curves are fitted into each cross-section. The graph with number of control polygon vertices on X-axis and angle between three consecutive points are obtained. The graphs are analyzed for extracting the features. Results are shown. The parameterized feature model of object is created

[Fig. 5.6]. This model can be used to extract the features from point cloud of similar type of object.

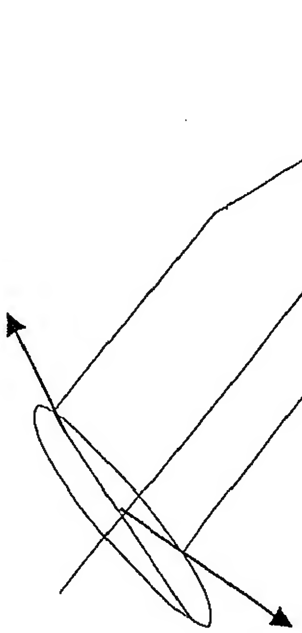


Fig 5.6: Parameterized feature model developed to extract the features

The process starts with aligning the point cloud with respect to the feature model. It is done using datum axis and datum plane of two models [Fig. 5.7]. After the alignment, point cloud is cut with the planes that contain the feature curves and feature points. The feature points and curves are extracted from the cloud as described in chapter 4. As explained in chapter 5, surface patches are developed. Post-processing is done on those surfaces to get the final CAD model [Fig. 5.8].

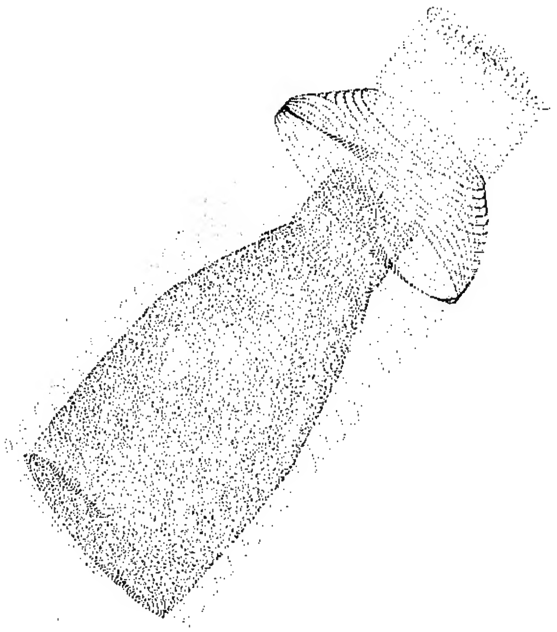


Fig 5.7: Aligned point cloud with the feature model

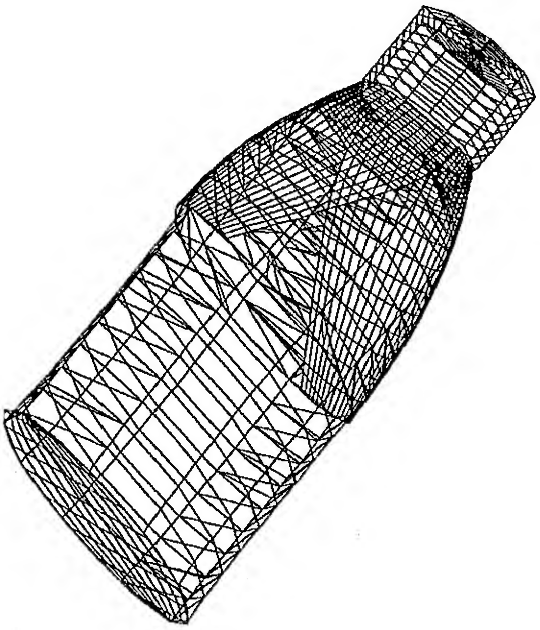


Fig 5.8: Final surface model after post-processing on surface patches

Chapter 6

CONCLUSIONS

6.1 SUMMARY

This work highlights various aspects of reverse engineering and feature-based design. The steps for surface creation in reverse engineering are explicitly described with the help of few examples, which shows that,

(1) Conventional reverse engineering methodology just concentrates on the geometry of the object and

(2) Embedded features are not recognized which can be used for certain downstream applications.

The principles of feature recognition technique, feature based design method and utility of development of user-defined features' libraries are elaborated.

The principle aim of this thesis is to integrate the techniques of reverse engineering and feature based design, so as to improve the process of solid model creation in terms of efficiency and accuracy. The process of integration is handled in two stages: feature extraction stage and feature based modeling stage.

The method developed to extract the features is automatic while the process of feature based modeling is interactive. The transition between these two stages requires user interaction. For feature extraction, the graphs between the angle, which is a measure of change in direction of control plot vectors of an interpolating B-spline and arc length of this curve, are generated for a variety of cases. Algorithms are developed so as to create templates for each curve feature. These algorithms are applied to curves generated in different cross-sections of given point cloud to extract the curve features. The results for different cross-sections are analyzed to extract the surface features. In feature based modeling stage, a four-layer structure network is proposed to describe each feature that is made of different surface features. A small library of features is developed which can be used to generate different CAD models. This methodology has been explained with the help of an example of solid model creation of a bottle, which embodies different features. Potential applications for this feature-based reverse engineering system include design of saddletrees in a saddlery and design of shoe soles in shoe industry.

6.2 SCOPE FOR FURTHER WORK

- The curve extraction process is directly affected by the quality of point cloud data, which depends upon the method of digitization. To obtain better results, it is necessary to have a better quality point cloud in terms of scatter. It can be achieved by using non-contact type of digitizers.
- The proposed method of parameterization is applicable to sweep surfaces, which can be extended to include complex surface features that are generated by methods, like lofting of curves.
- In the present work, the link among parameters of various features is missing and hence updating has to be done interactively. A data-structure can be developed so as to appropriately link parameters of various features so that automatic updating of the corresponding features is obtained.
- The module of feature based design and generation of surface models from library of features, which need user interaction at present, is to be automated and linked with feature extraction, to provide more powerful tool to the user.
- Through learning, the neural net can collect the characteristics of a geometric or topological pattern and recognize existing useful features. By using the templates for the features, one can teach neural nets to achieve faster execution speed, since execution speed is limited to simple mathematical computations and does not use either a search or logical rules to parse information.

REFERENCES

[1] Shah J J, Mantyla M, Nau D S, "Advances in Feature Based Manufacturing", *Elsevier Science B V.*, 1998.

[2] Hoffmann C M, Arinyo R J, "On User-defined Features", *Computer Aided Design*, Vol.30, No. 5, 1998, pp. 321-332.

[3] Sarkar B, Menq C-H, "Smooth–surface Approximation and Reverse Engineering", *Computer Aided Design*, Vol. 23, No. 9, 1991, pp. 623-628.

[4] Dobson G T, Waggenspack Jr. W N, Lamousin H J, "Feature Based Models for Anatomical Data Fitting", *Computer Aided Design*, Vol. 27, No. 2, 1995, pp. 139-146.

[5] Au C K, Yuen M M F, "Feature-based Reverse Engineering of Mannequin for Garment Design", *Computer Aided Design*, Vol. 31, 1999, pp. 751-759.

[6] Qamhiyah A Z, "A Strategy for the Construction of Customized Design Libraries for CAD", *Computer Aided Design*, Vol. 30, No. 11, 1998, pp. 897-904.

[7] Kanchana J, "Reverse Engineering", *4th SERC school on Advanced Manufacturing Technology*, May 18-30, 1998, PSG College of Technology, 1998.

[8] Bernard A, "Reverse Engineering for Rapid Product Development: A State of the Art", *Universite Nancy I, France*, 1999.

[9] Chivte P N, Jablokow A G, "Solid Model Generation from Measured Point Data", *Computer Aided Design*, Vol. 25, No. 9, 1993, pp. 587-600.

[10] Surfacer Version 8.0, "Imageware Surfacer User's Guide", *Imageware, Inc.* U.S.A., 1998.

APPENDIX

CASE STUDY [1]

In this project, CAD model of a composite shoe sole was to be developed. The requirement was to mix the features on the bottom side of one sole and the features on the top side of another sole. The sizes of two soles were different. The steps carried out were,

- Scanning both the models [Fig. A1].
- Scaling the point cloud of one sole to the dimensions of another one [Fig. A2].
- Curve fitting in both the clouds [Fig. A3].
- Registering the two data together to get a single curve network.
- Generating surfaces.
- Extending and trimming the surfaces to get the surface model.
- Post-processing of the surface model to get the final solid model [Fig. A4]

The trimmed surfaces were found to be problematic in obtaining the solid model from surface model.

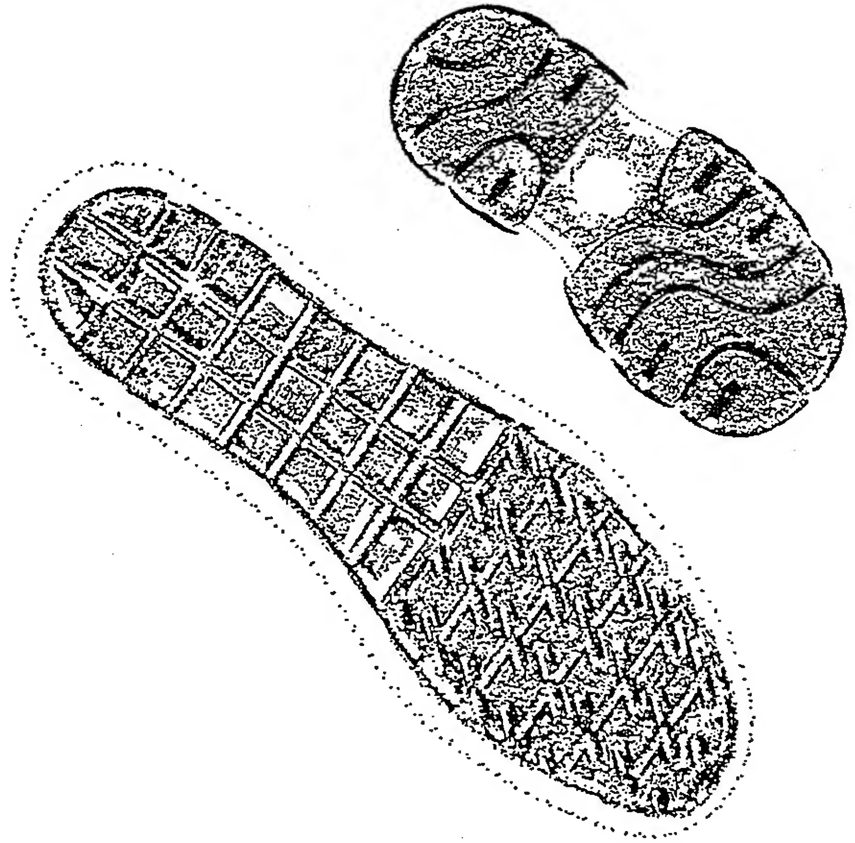


Fig. A1: Point Cloud of Two Soles in Original Size

0 2 4 6 8 in Scale: 1:2.2751

0 5 10 15 20 cm

Surfacer 8.1
CAD-P LAB
Printed: 03/13/00 03:02:26 AM
Project Name:
PartID: Rev:
By: Swapnil Chandsarkar
Date Effective:

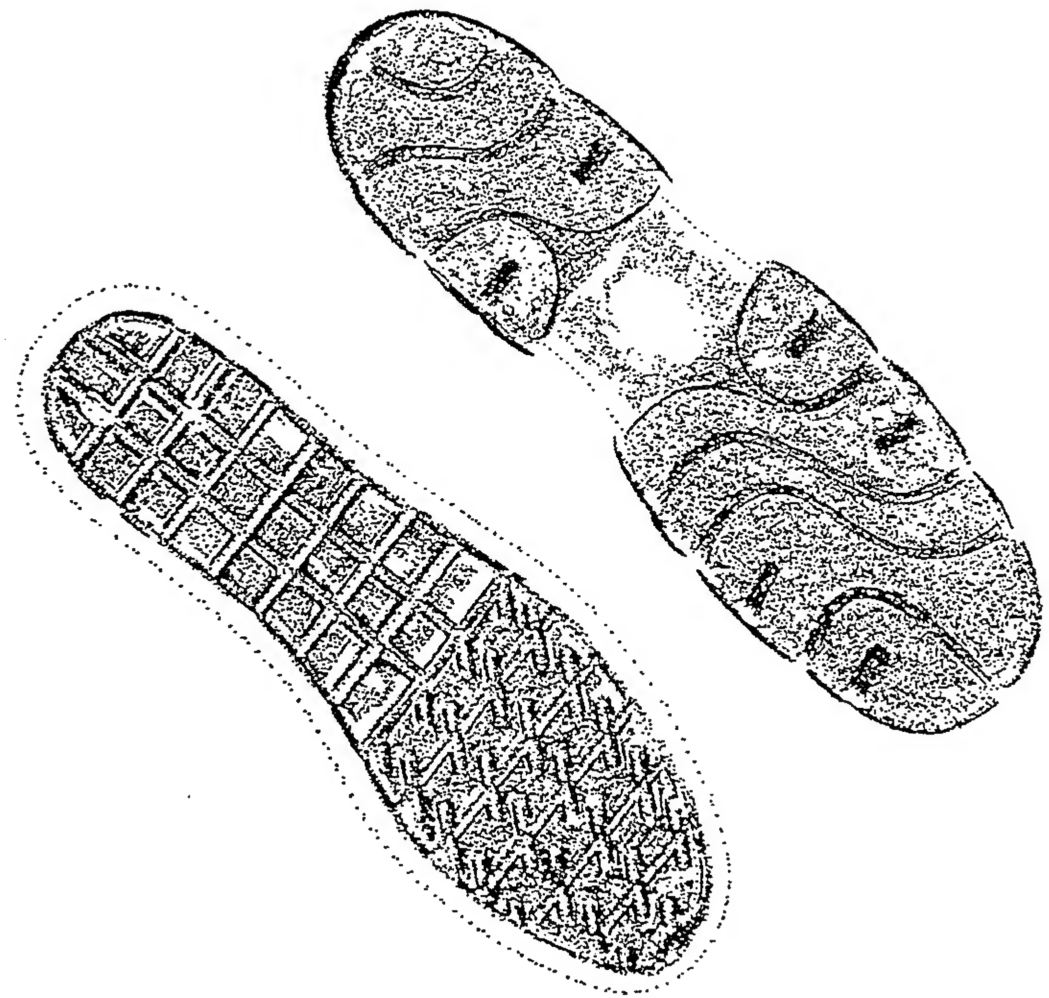


Fig. A2: Point Cloud of Two Soles in Same Size

Surfacer 8.1
CAD-P LAB
Printed: 03/13/00 04:06:04 AM
Project Name:
PartID: Rev:
By: Swapnil Chandsarkar
Date Effective:

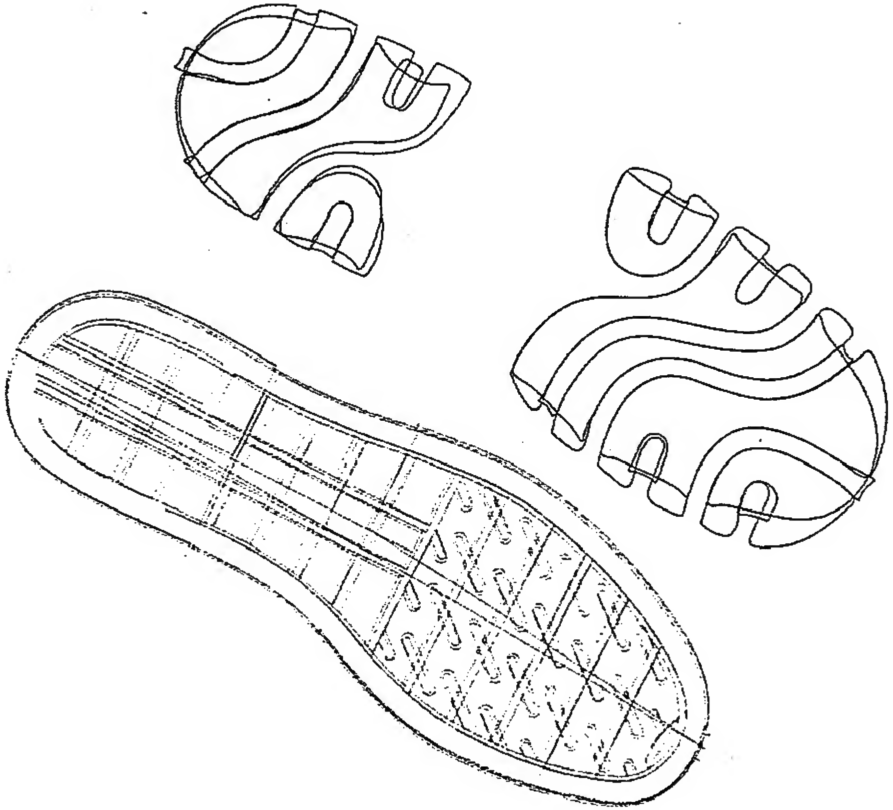


Fig. A3: Curves generated through both point clouds

0 2 4 6 8 10 in

0 5 10 15 20 25 cm

Scale: 1:2.617

Surfacer 8.1
CAD-P LAB
Printed: 03/13/00 03:17:17 AM
Project Name:
PartID: Rev:
By: Swapnil Chandsarkar
Date Effective:

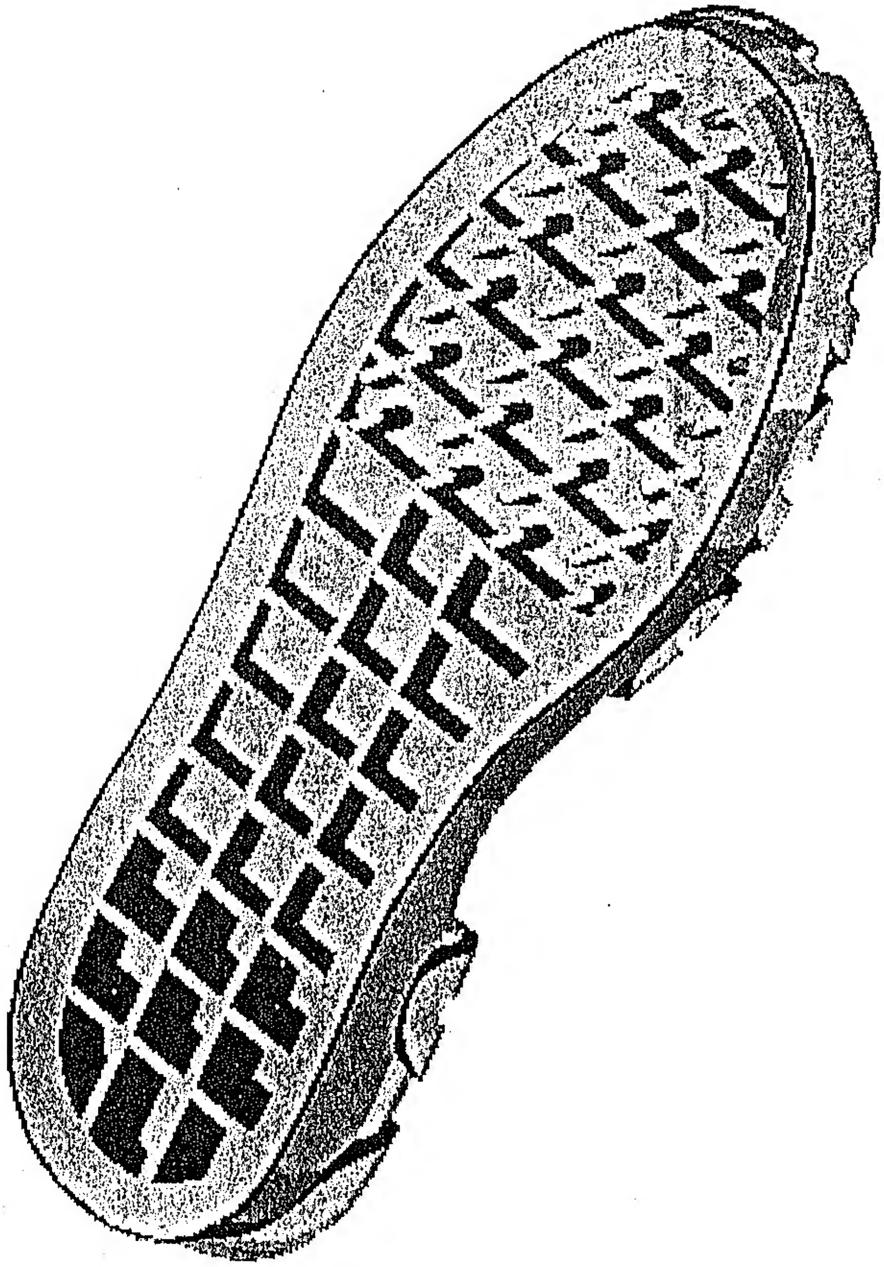


Fig. A4: Solid Model of Composite Sole

CASE STUDY [2]

The aim of this project was to generate the CAD model of a fan.

The steps involved in the project are described below.

- The physical model was scanned in multiple set-ups. For this purpose, the scanning work was divided in two parts. First the hub portion was scanned fully and then a single feature inside the hub was scanned. Similarly, two blades were scanned. Point data for both the blades were registered and error analysis was done to ensure that there is no variation in the profile of blades [Fig. A5].
- Curves were created through the point cloud of a single blade [Fig. A6].
- Surfaces were created with the help of these curves to get the surface model of a single blade.
- By making a polar array of a group of blade surfaces, surface model was obtained.
- Surfaces for the hub portion were generated in I-Deas (MS 6) software.
- All the surfaces were stitched together to get the final solid model. [Fig. A7].

The CAD model was to be used to create the NC code. The part was to be made in a two-part mold. Parting line, which was in a complex 3-D curve form, posed difficulty in making the final CAD model.

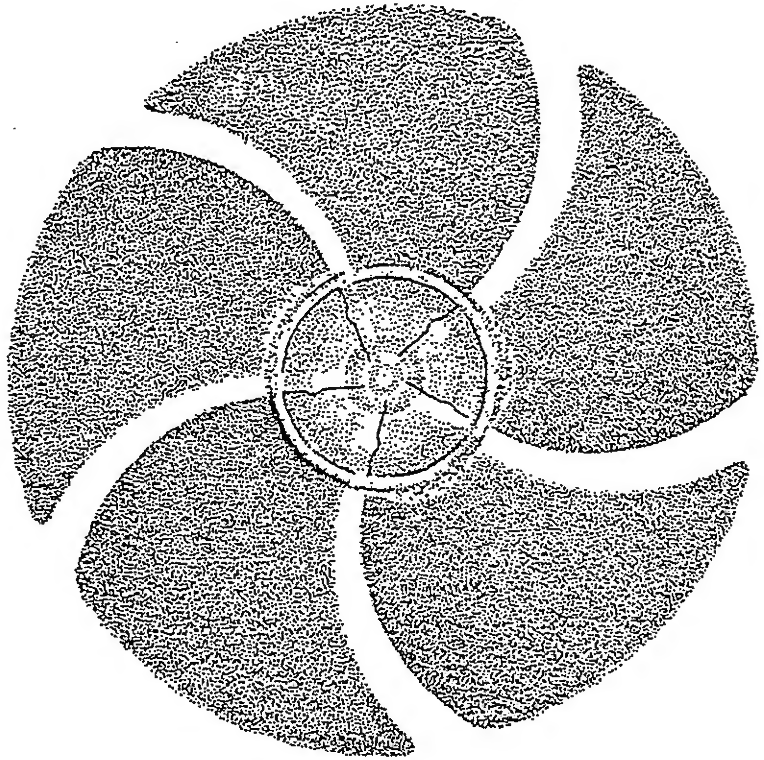


Fig. A5: Point Cloud of the Fan Model

Scale: 1:4.0596

Surfacer 8.1
CAD-P LAB
Printed: 03/13/00 02:27:48 AM
Project Name:
PartID: Rev:
By: Swapnil Chandsarkar
Date Effective:

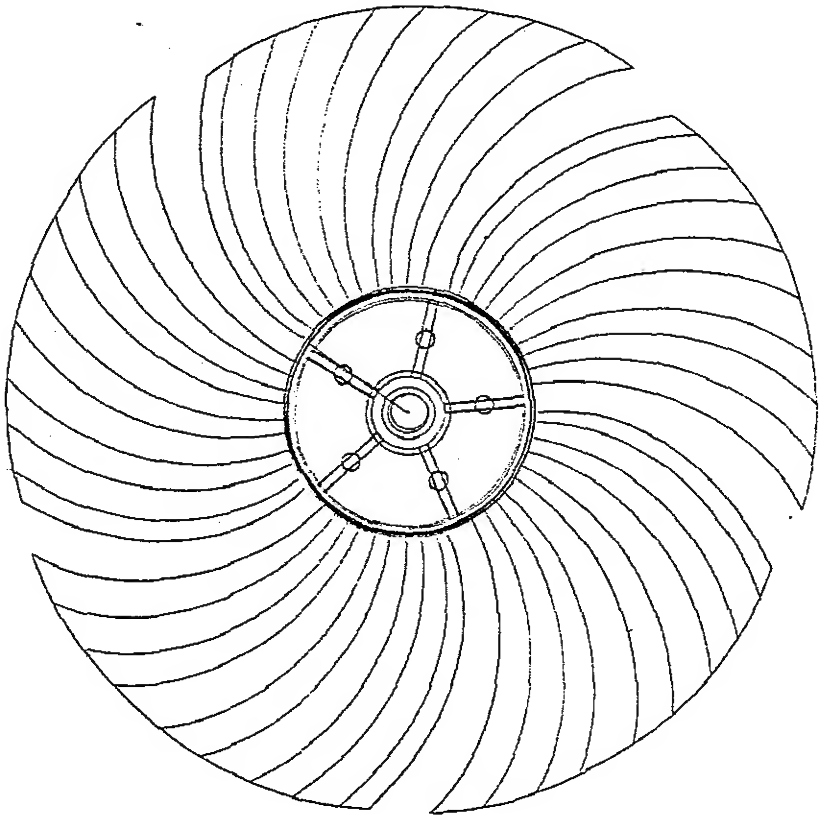


Fig. A6: Curves Generated Interactively through point cloud

Surfacer 8.1
CAD-P LAB
Printed: 03/13/00 02:29:54 AM
Project Name:
PartID: Rev:
By: Swapnil Chandsarkar
Date Effective:

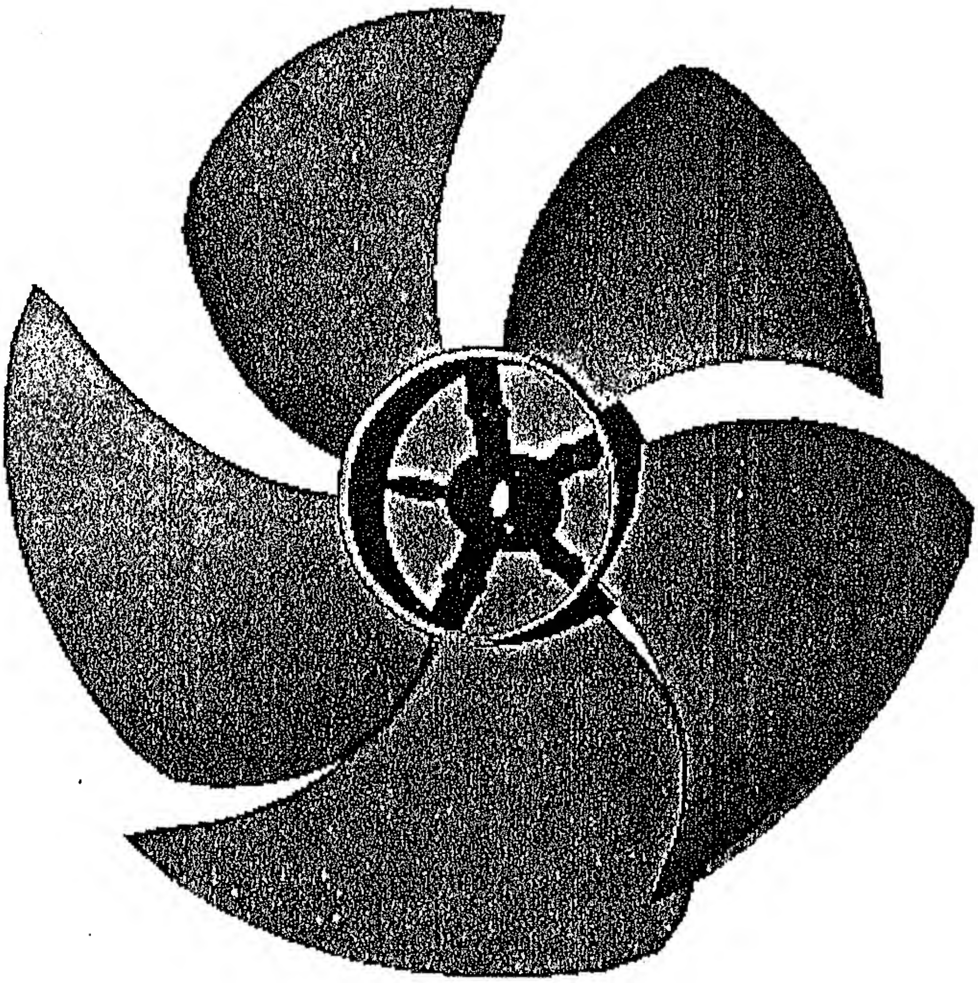


Fig. A7: Solid Model of the Fan